Livro do Pré-Simpósio
IX Symposium on Virtual and Augmented Reality
Petrópolis – RJ, 28 de Maio de 2007

Claudio Kirner
Robson Siscoutto
Editores

Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações

Livro do

Pré-Simpósio

IX Symposium on Virtual and Augmented Reality

Petrópolis – RJ, 28 de Maio de 2007.

Apresentação

Somos uma comunidade jovem, multidisciplinar, criativa, aberta a idéias e desafios, a qual atrai e acolhe novos talentos, sejam eles estudantes ou profissionais migrando de outras áreas. Foi pensando nesses novos e bem-vindos participantes do SVR que criamos o Pré-Simpósio (PS), apresentado pela primeira vez em São Paulo junto ao SVR 2004. O sucesso da iniciativa fez com que a Comissão Especial de Realidade Virtual da SBC, responsável pela organização e promoção do SVR, incluísse de forma definitiva o PS na programação de atividades do evento.

O principal objetivo do Pré-Simpósio é oferecer um curso rápido e abrangente sobre os principais conceitos e tecnologias das áreas de RV e RA, de tal forma a estabelecer um repertório básico que ajude o participante a melhor aproveitar tudo o que será exibido e discutido ao longo dos três dias de atividades principais do SVR. Criado, desenvolvido e apresentado por professores e pesquisadores seniores da comunidade de RV e RA, o Pré-Simpósio oferece aos participantes, além das 8 horas-aula, material complementar na forma de um texto abrangente que cobre os principais conceitos e tecnologias da área, cujo conteúdo vai muito além do que é apresentado ao vivo. No SVR 2004, o PS deu origem ao livro **“Realidade Virtual: Conceitos e Tecnologia”**. Esse livro, já esgotado, tem sido usado como referência em cursos técnicos e superiores, não só da área de computação e informática, mas também de design, comunicação e artes.

Após um processo de reestruturação e revisão da publicação do Pré-Simpósio do SVR 2004, bem como a ampliação e criação de novos capítulos, foi publicado, no SVR 2006, o livro denominado **“Fundamentos e Tecnologia de Realidade Virtual e Aumentada”**. Esse livro foi fruto do trabalho colaborativo de representantes de uma comunidade jovem e atuante, que está

crescendo em número e qualidade, podendo ser constatado em cada nova edição do Symposium on Virtual and Augmented Reality (SVR).

Com o intuito de continuar a democratização e a disseminação do conhecimento sobre RV e RA, o PS do SVR 2007, lança este livro denominado **"Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações"** contribuindo para a expansão desta comunidade. É assim, com grande prazer, que apresentamos o resultado do trabalho de 34 autores da comunidade brasileira de RV, que não mediram esforços para produzir este texto didático e de qualidade, e aos quais muito agradecemos.

Sem similares em língua portuguesa, estas obras dos Pré-Simpósios oferecem uma visão geral e abrangente de conceitos, tecnologias, arquiteturas e aplicações, constituindo-se em obras de referência para profissionais e pesquisadores, e útil porta de entrada para estudantes, iniciantes e profissionais de outras áreas do conhecimento interessados em ingressar no fascinante mundo da tecnologia de realidade virtual e aumentada. Esperamos que você aprecie e aproveite bastante este trabalho. Divirta-se.

Claudio Kirner[1] e Robson Siscoutto[2]

EDITORES

[1] ckirner@unimep.br
[2] robson.siscoutto@poli.usp.br

Coordenação de Produção e Editoração:
Robson Augusto Siscoutto - UNIC

Criação da Capa:
Emanuel Santana - Faculdade de Comunicação Social
UNIAGÊNCIA - UNIC – (65) 8411-4787

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro)

Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações
Claudio Kirner, Robson Siscoutto - editores. – Petrópolis – RJ,
Editora SBC – Sociedade Brasileira de Computação, Porto Alegre, 2007.
“Livro do pré-simpósio, IX Symposium on Virtual and Augmented Reality”

Bibliografia.
1. Realidade Virtual, Realidade Aumentada I. Kirner, Claudio II. Siscoutto, Robson.

ISBN 85-7669-108-6

Índice para catálogo sistemático:
1. Realidade Virtual e Aumentada: Ciência da Computação 006

Este livro foi especialmente editado, em tiragem limitada, a partir de conteúdos desenvolvidos para o curso “Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações” apresentado no pré-simpósio, em 28 de Maio de 2007, associado ao IX *Symposium on Virtual and Augmented Reality*, realizado em Petrópolis – RJ de 28 a 31 de Maio de 2007, promovido pela Sociedade Brasileira de Computação e organizado pelo Laboratório Nacional de Computação Científica (LNCC).

Petrópolis – RJ
2007

Sumário

SEÇÃO A

CONCEITOS

Capítulo 1

Fundamentos de Realidade Virtual e Aumentada

Claudio Kirner[1,2] e Robson Augusto Siscoutto[3]

[1] Faculdade de Ciências Exatas e da Natureza – Universidade Metodista de Piracicaba (UNIMEP)
Rodovia do Açúcar, Km 156 – 13400-911 – Piracicaba - SP – Brasil

[2] Centro Universitário Adventista de São Paulo (UNASP)
Est. de Itapecerica, 5859 CEP 05858-001 - São Paulo-SP - Brasil

ckirner@unimep.br

[3] Faculdade de Ciência e Tecnologia – Universidade de Cuiabá (UNIC)
Av. Beira Rio, 3100 – 78015-480 – Jardim Europa - Cuiabá - MT – Brasil

robson@unic.br

Abstract

Virtual reality and augmented reality are areas related to new user interface generation, making easy and improving the user interaction with computational applications. This chapter presents the concepts of virtual and augmented reality, showing the aspects of technology, design and application to be addressed in this book.

Resumo

Realidade Virtual e Realidade Aumentada são duas áreas relacionadas com as novas gerações de interface do usuário, facilitando e potencializando as interações do usuário com as aplicações computacionais. Este capítulo apresenta os conceitos de realidade virtual e aumentada, mostrando os aspectos de tecnologia, projeto e aplicação a serem abordados neste livro.

1.1. Introdução

Antes do advento do computador eletrônico, com o desenvolvimento do ENIAC em 1945, as pessoas utilizavam interfaces naturais para interagir com o mundo no seu dia-a-dia, usando seus sentidos. Em raras ocasiões, elas necessitavam interagir com máquinas, apertando botões ou acionando alavancas.

O computador eletrônico trouxe um novo processo sofisticado de interação com as aplicações, exigindo conhecimento simbólico (abstrato) e necessidade de treinamento, uma vez que o conhecimento do mundo real já não era suficiente.

Apesar dos benefícios da tecnologia, a sofisticação das interfaces do usuário fez com que as pessoas tivessem que se ajustar às máquinas, durante muitas décadas. Felizmente, os pesquisadores, desde o início da era dos computadores, buscaram maneiras de fazer com que as máquinas se ajustassem às pessoas, o que foi se conseguindo com a evolução das tecnologias de hardware, software e telecomunicações. Surgiram então interfaces de voz, interfaces tangíveis, interfaces hápticas, etc, possibilitando, aos usuários, acessarem aplicações como se estivessem atuando no mundo real, falando, pegando, apertando, fazendo gestos, etc.

Há vários pontos importantes nessa evolução, envolvendo invenções como: computador eletrônico à válvula, transistor,

circuito integrado, monitor de computador, computação gráfica, rede, computador pessoal (PC), Internet, Web, software livre, além de outros inventos tecnológicos, cuja integração permitiram o uso transparente da tecnologia, por parte do usuário. O usuário já não precisa perceber a presença da tecnologia, pois ela trabalha para ele de forma invisível em qualquer lugar, dando origem aos termos: ubíquo e pervasivo.

Nesse contexto, a realidade virtual [Burdea, 1994] e a realidade aumentada [Azuma, 1997] têm um papel importante, pois são interfaces computacionais avançadas, que ainda não foram implantadas de forma expressiva na sociedade.

As primeiras interfaces computacionais, usadas nas décadas de 40 e 50, eram baseadas em chaves e lâmpadas, que permitiam uma comunicação com o computador baseada em linguagem de máquina. Na década de 60, surgiram as consoles com vídeo, dando início às interfaces gráficas rudimentares. Com a utilização de microprocessadores, nas décadas de 70 e 80, os microcomputadores se popularizaram, usando interface baseada em comando, como o DOS. A evolução desta interface resultou no Windows, que, explorando técnicas de multimídia, persiste até hoje. Apesar de interessante e de ter bom potencial de uso, a interface Windows fica restrita à limitação da tela do monitor e ao uso de representações como menus e ícones.

A realidade virtual surge então como uma nova geração de interface, na medida em que, usando representações tridimensionais mais próximas da realidade do usuário, permite romper a barreira da tela, além de possibilitar interações mais naturais. A realidade virtual teve suas origens na década de 60, com o desenvolvimento do ScketchPad por Ivan Sutherland [Sutherland, 1963], mas só ganhou força na década de 90, quando o avanço tecnológico propiciou condições para a execução da computação

gráfica interativa em tempo real. Apesar das vantagens da realidade virtual, ela necessitava de equipamentos especiais como capacete, luva, óculos estereoscópicos, mouses 3D, etc, para fazer com que o usuário fosse transportado para o espaço da aplicação, onde realiza suas interações. Além disso, o "transporte" do usuário para o ambiente virtual (desconhecido) causava um desconforto inicial e dificuldades de interação, exigindo, muitas vezes, treinamento. Esses problemas inibiram a popularização da realidade virtual como uma nova interface do usuário.

Por outro lado, a evolução tecnológica também propiciou , na década de 90, o aparecimento da realidade aumentada, permitindo a sobreposição de objetos e ambientes virtuais com o ambiente físico, através de algum dispositivo tecnológico. Essas aplicações ficaram mais acessíveis somente no início dos anos 2000, com a convergência de técnicas de visão computacional, software e dispositivos com melhor índice de custo-benefício. Além disso, o fato dos objetos virtuais serem trazidos para o espaço físico do usuário (por sobreposição) permitiu interações tangíveis mais fáceis e naturais, sem o uso de equipamentos especiais. Por isso, a realidade aumentada vem sendo considerada uma possibilidade concreta de vir a ser a próxima geração de interface popular, a ser usada nas mais variadas aplicações em espaços internos e externos.

Enquanto a realidade virtual depende de equipamentos de visualização, como monitor, projetor e capacete, normalmente utilizados em ambientes fechados, a realidade aumentada não apresenta esta restrição com dispositivos misturadores, podendo ser usada em qualquer ambiente (fechado ou aberto), sendo, portanto mais abrangente e universal. Por outro lado, tanto a realidade virtual quanto a realidade aumentada podem ser usadas em aplicações individuais e em aplicações coletivas locais ou remotas, propiciando experiências colaborativas [Billinghurst, 1999]

[Benford, 1998; Kirner, 2004]. No entanto, a realidade aumentada apresenta a vantagem de permitir o uso de ações tangíveis [Kawashima, 2001] e de operações multimodais, envolvendo voz, gestos, tato, etc, facilitando o trabalho do usuário sem a necessidade de treinamento.

Assim, a convergência tecnológica e o desenvolvimento de interfaces estão apontando para a nova geração de interfaces computacionais baseadas em realidade aumentada para uso nas mais variadas áreas, desde entretenimento, como jogos, até experimentos científicos coletivos, constituindo verdadeiros laboratórios de pesquisa.

1.2. Conceitos e Definições

Em função da abundância de termos e de interesses das áreas de realidade virtual e aumentada, em função de sua multidisciplinaridade, serão abordados em seguida alguns conceitos e definições envolvidos com o assunto.

1.2.1. Multimídia

Multimídia consiste na integração, controlada por computador, de textos gráficos, imagens, vídeo, animações, áudio e outras mídias, que possam representar, armazenar, transmitir e processar informações de forma digital [Marshal, 2001].

Aplicações multimídia são potentes e simples de usar, mas restringem a visualização do usuário à tela do computador (2D). Esta deficiência pode ser atenuada com o aproveitamento do espaço da tela do monitor, através de múltiplas janelas sobrepostas ou espalhadas.

1.2.2. Realidade Virtual

A Realidade Virtual (RV) é uma "interface avançada do usuário" para acessar aplicações executadas no computador, propiciando a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador. O sentido da visão costuma ser preponderante em aplicações de realidade virtual, mas os outros sentidos, como tato, audição, etc. também podem ser usados para enriquecer a experiência do usuário.

A modelagem dos ambientes virtuais, usando linguagens como VRML (*Virtual Reality Modeling Language*) [VRML97, 2007] e sua sucessora, X3D [Walsh, 2001; Web3d, 2007], além de outras linguagens e ferramentas de autoria, permite, ao usuário, visualizar ambientes tridimensionais, movimentar-se dentro deles e manipular seus objetos virtuais. Os objetos virtuais podem ser animados, apresentando comportamentos autônomos ou disparados por eventos. A Figura 1.1 mostra um exemplo de ambiente virtual visto no monitor.

Figura 1.1. Exemplo de aplicação de realidade virtual.

A interação do usuário com o ambiente virtual é um dos aspectos importantes da interface e está relacionada com a capacidade do computador detectar e reagir às ações do usuário, promovendo alterações na aplicação [Bowman, 2005]. O usuário, interagindo com um ambiente virtual tridimensional realista, em

tempo-real, vendo as cenas serem alteradas como resposta aos seus comandos, como ocorre nos videogames atuais, torna a interação mais rica e natural, gerando mais engajamento e eficiência.

Nos ambientes virtuais, a interação mais simples é a navegação, decorrente da movimentação do usuário no espaço tridimensional, através de algum dispositivo, como o mouse 3D, comandos de voz ou de gestos detectados por algum dispositivo de captura, resultando na visualização de novos pontos de vista do cenário. Nesse caso, não há mudanças no ambiente virtual, mas somente um passeio exploratório. Interações, propriamente ditas, com alterações no ambiente virtual, ocorrem quando o usuário entra no espaço virtual das aplicações e visualiza, explora, manipula e aciona ou altera os objetos virtuais, usando seus sentidos, incluindo os movimentos tridimensionais de translação e rotação naturais do corpo humano.

A interface baseada em realidade virtual permite que habilidades e conhecimento intuitivos do usuário possam ser utilizados para a manipulação dos objetos virtuais. Esse tipo de interação é realizado, através de dispositivos não convencionais, como capacete de visualização ou luvas, o próprio corpo, como gestos e comandos de voz, ou até mesmo dispositivos convencionais como mouse, teclado e monitor de vídeo. O usuário deve ter a impressão de estar atuando dentro do ambiente virtual, apontando, pegando, manipulando e executando outras ações sobre os objetos virtuais, em tempo-real. Normalmente, os atrasos admissíveis para que o ser humano tenha a sensação de interação em tempo-real estão em torno de 100 milisegundos, tanto para a visão, quanto para as reações de tato, força e audição. Isto impõe um compromisso do sistema (processadores, software, dispositivos, complexidade do ambiente virtual, tipo de interação, etc) em funcionar com taxas mínimas de 10 quadros por segundo na renderização das imagens (sendo desejado algo em torno de 20

quadros por segundo para suportar melhor as cenas animadas) e de 100 milisegundos de atraso nas reações aos comandos do usuário. Assim, a complexidade do mundo virtual, os dispositivos usados, o software e a configuração do sistema devem ser ajustados para funcionar com as taxas mínimas de renderização e reação.

Existem muitas definições de realidade virtual, envolvendo diversos aspectos [Burdea, 1994; Vince, 1995, 2004; Kirner, 1996; Sherman, 2003]. Uma definição, sintetizando as várias considerando as discussões apresentadas até agora, é a seguinte:

“Realidade virtual é uma interface avançada para aplicações computacionais, que permite ao usuário navegar e interagir, em tempo real, com um ambiente tridimensional gerado por computador, usando dispositivos multisensoriais”.

Apesar da realidade virtual também usar múltiplas mídias, ela enfatiza a interação do usuário com o ambiente tridimensional e a geração das imagens em tempo real. Para que isso ocorra, a plataforma computacional deve ser apropriada para aplicações de realidade virtual, apresentando boa capacidade de processamento gráfico para a renderização de modelos tridimensionais em tempo real, e suportando dispositivos não convencionais de interação para atender à demanda multisensorial.

A comparação entre multimídia e realidade virtual pode ser vista da seguinte maneira:

- Multimídia envolve imagens capturadas ou pré-processadas; prioriza a qualidade das imagens; exige alta capacidade de transmissão; usa técnicas de compressão de dados; atua no espaço 2D; e funciona com dispositivos convencionais.

- Realidade virtual envolve imagens calculadas em tempo real; prioriza a interação com o usuário; exige alta capacidade de

processamento; usa técnicas e recursos de renderização de modelos tridimensionais e funciona com dispositivos especiais.

Tanto na multimídia, como na realidade virtual, o usuário tem de ser transportado para o domínio da aplicação (ambiente virtual), podendo causar-lhe desconforto frente ao desconhecido, além da necessidade de adaptação e treinamento.

1.2.3. Realidade Aumentada

A realidade aumentada é definida de várias maneiras:

a) é o enriquecimento do ambiente real com objetos virtuais, usando algum dispositivo tecnológico, funcionando em tempo real;

b) é uma melhoria do mundo real com textos, imagens e objetos virtuais, gerados por computador [Insley, 2003];

c) é a mistura de mundos reais e virtuais em algum ponto da realidade/virtualidade contínua, que conecta ambientes completamente reais a ambientes completamente virtuais [Milgran, 1994];

d) é um sistema que suplementa o mundo real com objetos virtuais gerados por computador, parecendo coexistir no mesmo espaço e apresentando as seguintes propriedades:

- combina objetos reais e virtuais no ambiente real;
- executa interativamente em tempo real;
- alinha objetos reais e virtuais entre si;
- aplica-se a todos os sentidos, incluindo audição, tato e força e cheiro [Azuma, 2001].

A Figura 1.2 apresenta um exemplo de aplicação de realidade aumentada com uma mesa real enriquecida com vaso e carro virtuais.

Figura 1.2. Realidade aumentada com vaso e carro virtuais sobre a mesa.

Essa tecnologia deverá ter grande impacto no relacionamento das pessoas, através de novas maneiras de realizar visualização, comunicação e interação com pessoas e informação.

A realidade aumentada e a realidade virtual [Bimber, 2004] podem ser comparadas da seguinte forma:

- a realidade aumentada enriquece a cena do mundo real com objetos virtuais, enquanto a realidade virtual é totalmente gerada por computador;

- no ambiente de realidade aumentada, o usuário mantém o sentido de presença no mundo real, enquanto que, na realidade virtual, a sensação visual é controlada pelo sistema;

- a realidade aumentada precisa de um mecanismo para combinar o real e o virtual, enquanto que a realidade virtual precisa de um mecanismo para integrar o usuário ao mundo virtual.

1.2.4. Hiper-realidade

O próximo passo da evolução das interfaces é incrementar a combinação do mundo real com o mundo virtual, através de novos

elementos e comportamentos para facilitar e potencializar a interação do usuário com os recursos que ele necessita no dia a dia.

Nesse contexto, surge o conceito de hiper-realidade [Tiffin, 2001], cuja definição é a seguinte: "hiper-realidade é a capacidade tecnológica de combinar realidade virtual, realidade física, inteligência artificial e inteligência humana, integrando-as de forma natural para acesso do usuário".

Ambientes de hiper-realidade permitirão que habitantes reais interajam com habitantes remotamente localizados, bem como com objetos ou formas de vida imaginárias ou artificiais, gerados por computador, em um mundo misturado. Esse mundo será formado por pessoas, animais, insetos, plantas, terrenos, construções e objetos virtuais inteligentes, todos integrados. Com a visão do mundo misturado, cada usuário poderá enxergar o que lhe interessa, de acordo com seu perfil ou sua necessidade, e interagir com os objetos, de forma a ter suas necessidades satisfeitas. Como exemplo, o usuário, ao caminhar ou dirigir seu automóvel por uma cidade (usando um capacete de visão óptica direta), poderá fazer solicitações por comandos de voz e ver legendas virtuais nos prédios e ruas orientando-o ou mostrando opções como: o melhor caminho para chegar a um destino; restaurantes de determinados tipos ou padrões; entretenimentos específicos; lojas; supermercados; hospitais; e assim por diante.

Muito do que se desenvolveu na Internet para facilitar a vida do usuário, poderá ser transportado para o mundo misturado de forma gráfica e seletiva. Assim, nesse mundo misturado com hiper-realidade, as pessoas deverão satisfazer muitas de suas necessidades, atuando num ambiente integrado inteligente, sendo atendidas de forma explícita ou implícita.

1.2.5. Rastreamento

O rastreamento, em ambientes de realidade virtual e aumentada, tem a função de identificar a posição da mão, da cabeça, do próprio usuário ou de algo atrelado a ele, como uma placa. Com isto, o sistema permite que o usuário exerça um controle de posicionamento em ambientes virtuais ou aumentados, podendo, por exemplo, movimentar-se e tocar, agarrar, mover e soltar objetos virtuais.

Para uso em aplicações de realidade virtual, muitos dispositivos de rastreamento foram desenvolvidos, usando princípios mecânicos, magnéticos, de ultrasom, etc. Cada tipo apresenta vantagens e desvantagens, mas em geral são caros.

Mais recentemente, com a popularização da webcam e com o avanço das técnicas de visão computacional e do poder de processamento dos microcomputadores, o rastreamento óptico passou a ser uma realidade, em função da disponibilidade e do baixo custo. A biblioteca ARToolKit [Lamb, 2007], usada em aplicações de realidade aumentada, utiliza esse tipo de rastreamento.

Assim, o capítulo 2 deste livro explora, em detalhes, os conceitos, tecnologia e uso de rastreamento óptico para aplicações de realidade virtual e aumentada.

1.2.6. Interação

A interação consiste na capacidade do usuário atuar em ambientes virtuais e aumentados, promovendo alterações e reações às suas ações. Esta é a principal característica dos jogos por computador, sendo o fator determinante para o envolvimento do usuário e o sucesso da aplicação.

Para que uma interação tenha efeito, é necessário um controle de posicionamento do usuário (rastreamento) e outros atributos do sistema como: apontamento e seleção de objetos, ativação de ações, etc.

O capítulo 3 deste livro trata dos detalhes da interação em ambientes e realidade virtual e aumentada, mostrando conceitos, implementação e exemplos.

1.3. Tecnologia, Projeto e Aplicações

As áreas de realidade virtual e realidade aumentada, assim como outras áreas, só se desenvolveram com a convergência de uma série de fatores, incluindo as pesquisas, a tecnologia, a disponibilidade de produtos e os custos acessíveis. Nesse sentido, tiveram papel fundamental os pesquisadores que disseminaram essas áreas com o desenvolvimento de recursos disponibilizados à sociedade gratuitamente. A linguagem VRML - *Virtual Reality Modeling Language* [VRML97, 2007] para uso em realidade virtual e a biblioteca ARToolKit [Lamb, 2007] para realidade aumentada são os exemplos mais marcantes de recursos gratuitos e livres, mas existem vários outros disponibilizados por pesquisadores e, mais recentemente, por empresas.

Em seguida, o leitor deverá ter uma visão geral sobre as seções deste livro, envolvendo os aspectos de tecnologia, projeto e aplicações.

1.3.1. Ambientes de Hardware e Software

O hardware de um computador é o núcleo de qualquer aplicação, oferecendo os recursos a serem usados. Suas características, ajustadas às necessidades da aplicação, são responsáveis pela obtenção do melhor índice de custo-benefício.

O hardware envolve muitos elementos, consistindo de: processador; placas especiais, como placas gráficas e sonoras; periféricos, incluindo dispositivos especiais multisensoriais; e infraestrutura de rede.

Para fazer o hardware funcionar adequadamente, são usados ambientes de software, denominados núcleos ou sistemas operacionais, que constituem máquinas virtuais poderosas encarregadas de suportar as necessidades de aplicações específicas de realidade virtual e aumentada.

O capítulo 4 deste livro aborda as questões de hardware e software básicos, mostrando suas características, exemplos e potencial de utilização.

1.3.2. Desenvolvimento e Avaliação de Sistemas

O processo de desenvolvimento de aplicações de realidade virtual e aumentada requer a utilização de técnicas de engenharia de software adaptadas ao contexto dessas áreas.

Além das técnicas de desenvolvimento, é importante usar técnicas de avaliação das aplicações, no sentido de aferí-las sob pontos de vista específicos.

O capítulo 5 deste livro trata desse assunto para a área de realidade virtual, servindo como ponto de partida para o processo de desenvolvimento e avaliação de aplicações de realidade aumentada.

1.3.3. Ferramentas

O desenvolvimento de aplicações de realidade virtual e aumentada pode ser facilitado, quando são usadas ferramentas apropriadas para cada caso, como linguagens, bibliotecas, ambientes visuais de

desenvolvimento, etc, constituindo ferramentas de autoria das aplicações.

A escolha certa das ferramentas depende de um bom conhecimento do domínio da aplicação e da disponibilidade de recursos de hardware e de software, além de recursos financeiros, quando se tratar de ferramentas comerciais.

O capítulo 6 deste livro faz uma análise de ferramentas para desenvolvimento de aplicações de realidade virtual e aumentada, citando exemplos e potencialidades de uso.

1.3.4. Sistemas Distribuídos

Aplicações simples de realidade virtual e aumentada costumam ser voltadas a um único usuário. No entanto, a demanda por trabalho colaborativo, envolvendo vários usuários remotos, vem crescendo e exigindo o desenvolvimento de aplicações sobre redes de computadores ou constituindo sistemas distribuídos.

Em função dos requisitos específicos das aplicações de realidade virtual e aumentada, é importante que os ambientes de rede sejam apropriados para dar-lhes suporte, fornecendo o comportamento esperado pelos usuários.

O capítulo 7 deste livro trata das questões de sistemas distribuídos de realidade virtual e aumentada, considerando seus requisitos e mostrando exemplos, aplicações e ferramentas.

1.3.5. Dispositivos Hápticos

Quando se trabalha com ambientes de realidade virtual e aumentada, manipulando objetos virtuais, sente-se falta da sensação de toque (háptica) que impõe mais realismo às aplicações. Em alguns casos, a sensação háptica, envolvendo tato e força, pode

até ser dispensada, mas, em outros, ela é essencial, como ocorre em situações de treinamento.

O treinamento médico é um exemplo típico, no qual o cirurgião deve ter as mesmas sensações de toque nos tecidos virtuais e nos tecidos reais. Para isto, são usados os dispositivos hápticos e os modelos geométricos deformáveis para se ajustarem às várias situações de pressão e ruptura.

O capítulo 8 deste livro discute os aspectos dos dispositivos hápticos e sua utilização em aplicações de realidade virtual e aumentada.

1.3.6. Ambientes Colaborativos

A evolução das tecnologias de informática e de telecomunicações vem propiciando a computação ubíqua ou pervasiva, que faz com que as aplicações computacionais estejam em vários lugares ao mesmo tempo. Isto ocorre com telefones celulares, Internet e outras aplicações.

Como resultado dessas facilidades, as aplicações colaborativas estão tendo um grande desenvolvimento, principalmente em educação, treinamento, projeto e entretenimento, demandando ambientes mais realistas e potencializados como ocorre com realidade virtual e aumentada.

O capítulo 9 deste livro aborda os conceitos, características, exemplos e potencialidades de ambientes colaborativos de realidade virtual e aumentada.

1.3.7. Jogos e Entretenimento

Jogos e entretenimento sempre tiveram um apelo muito grande nas pessoas. Com o surgimento da realidade virtual e da realidade aumentada, usando cenários e comportamentos mais realistas,

recursos dessas áreas passaram a ser usados para o desenvolvimento de novas aplicações ou para a remodelagem de aplicações já existentes.

O capítulo 10 deste livro trata do uso de realidade virtual e aumentada em jogos e entretenimento, mostrando suas características, exemplos e tendências.

1.3.8. Aplicações Médicas

A medicina é uma das áreas que mais demandaram o uso de realidade virtual e aumentada em educação, treinamento, diagnóstico, tratamento e simulação de cirurgia. Realidade virtual e realidade aumentada, pelas suas características de visualização 3D e de interação em tempo real, permitem a realização de aplicações médicas inovadoras, que antes não podiam ser realizadas.

O capítulo 11 deste livro discute como a realidade virtual e a realidade aumentada podem ser usadas em aplicações médicas, mostrando exemplos e explorando os recurso de visualização e de sensações multisensoriais em situações físicas e psicológicas.

1.3.9. Visualização

As imagens sempre foram usadas para melhorar as condições de visualização de informação das pessoas, através de gráficos e figuras.

Por outro lado, ambientes de realidade virtual e aumentada amplificam as capacidades das pessoas em avaliarem informações tridimensionais, na medida em que flexibilizam a atuação no espaço 3D e permitem o uso de interações multimodais, possibilitando maior riqueza de detalhes, melhores técnicas de interação e mais desempenho.

O capítulo 12 deste livro aborda as questões de visualização apoiada por realidade virtual e aumentada, mostrando exemplos e tendências.

1.4. Conclusões

Realidade virtual e realidade aumentada são áreas recentes do conhecimento que vêm dando, aos usuários, melhores condições de interação com aplicações computacionais, propiciando a eles interações naturais e potencialização de suas capacidades.

Para isso, muitos recursos são utilizados, envolvendo hardware, software, periféricos, redes, tecnologias especiais, técnicas de projeto e avaliação e o desenvolvimento de aplicações.

Este livro agrupa várias questões envolvidas com o uso de realidade virtual e realidade aumentada para facilitar a vida das pessoas, considerando os conceitos e tendências, assim como os aspectos de tecnologia, projeto e aplicações.

Este capítulo teve como objetivo dar uma visão conceitual de realidade virtual e aumentada, incluindo tendências, além de apresentar um resumo do que o leitor irá encontrar na sequência de leitura.

1.5. Referências Bibliográficas

Azuma, R. (1997) “A Survey of Augmented Reality", Presence: Teleoperators and Virtual Environments, v .6, n.4, August, p. 355-385.

Azuma, R. et al. (2001) “Recent Advances in Augmented Reality.” IEEE Computer Graphics and Applications, v .21, n.6, p. 34-47.

Benford, S. et. al. (1998) "Understanding and Constructing Shared Spaces with Mixed Reality Boundaries". ACM ToCHI, v.5, N.3, p. 185-223.

Billinghurst, M., Kato, H. (1999) "Collaborative Mixed Reality", Proc. of the International Symposium on Mixed Reality, ISMR'99, Springer -Verlag, p. 261-284.

Bimber, O., (2004) "Augmented Reality - Part 1 - Introduction and Overview" <http://www.uni-weimar.de/~bimber/Pub/AR/>

Bowman, D., et al. (2005). *"3D User Interfaces: Theory and Practice"*. Boston, MA: Addison-Wesley.

Burdea, G., Coiffet,P. (1994) "Virtual RealityTechnology", John Wiley & Sons.

Insley, S. (2003) "Obstacles to General Purpose Augmented Reality" <http://islab.oregonstate.edu/koc/ece399/f03/final/insley2.pdf>

Kawashima, T. et. al. (2001) "Magic Paddle: A Tangible Augmented Reality Interface for Object Manipulation", Proc. of ISMR2001, p. 194-195.

Kirner, C., Pinho, M.S. (1996) "Introdução a Realidade Virtual". Mini-Curso, JAI/SBC, Recife, PE.

Kirner, C. (2004) "Mãos Colaborativas em Ambientes de Realidade Misturada" Anais do 1º Workshop de Realidade Aumentada, Piracicaba, SP, p. 1-4.

Lamb,P. (2007) "ArtoolKit" <www.hitl.washington.edu/artoolkit/>

Marshall, D. (2001) "What is Multimedia?" http://www.cs.cf.ac.uk/Dave/Multimedia/node10.html

Milgram, P. et. al. (1994) "Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum". Telemanipulator and Telepresence Technologies, SPIE, V.2351, p. 282-292.

Sherman, W.R., Craig, A.B. (2003) “Understanding Virtual Reality”, Morgan Kaufmann.

Sutherland, I. E. (1963) “SKETCHPAD: A MAN-MACHINE GRAPHICAL COMMUNICATION SYSTEM” AFIPS Conference Proceedings, Spring Joint Computer Conference, Detroit <faculty.cs.tamu.edu/hammond/courses/SR/papers/Sutherland/Sutherland1963Sketchpad.pdf>

Tiffin, J., Terashima, N. ed. (2001) “Hyper-reality: Paradigm for the Third Millennium”. Routledge.

Vince, J. (1995) “Virtual Reality Systems”, Addison-Wesley.

Vince, J. (2004) “Introduction to Virtual Reality”, Springer-Verlag, 2nd edition.

VRML97 (2007). “The Virtual Reality Modeling Language”. <http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/>

Walsh, A.E., Bourges-Sévenier, M. (2001), “Core WEB3D”, Prentice Hall.

Web3D Consortium (2007) “X3D Documentation”. <http://www.web3d.org/x3d/>

Capítulo 2

Rastreamento Óptico para Sistemas de Realidade Virtual e Aumentada

Luís Augusto Consularo[1], Regina Célia Coelho[1] e Nivaldi Calonego Jr.[2]

[1] Universidade Metodista de Piracicaba (UNIMEP)
Rodovia do Açúcar, km 156 – 13400-911 – Piracicaba – SP
{consularo,rccoelho}@unimep.br

[2] Universidade Estadual do Mato Grosso (UNEMAT)
nivaldi.calonegojr@gmail.com

Abstract

The camera tracking is particularly useful for Virtual and Augmented Reality systems because tracked objects may serve as interaction devices in 3D scenes. Real object representations are built from the optical tracking of real object projections on photosensor arrays. Cameras as tracking devices have the advantage of not having mechanical contact. However, their effective application requires the knowledge of the main parameters and features involved in the tracking process. This chapter is a very short introduction to the camera tracking problem with an ARToolkit-based example for punctual objects.

Resumo

O rastreamento óptico é útil para sistemas de Realidade Virtual e Aumentada, pois objetos rastreados servem como dispositivos de interação em cenas 3D. Representações de objetos reais são construídas com o rastreamento de suas projeções sobre uma matriz de fotossensores. As câmeras têm a vantagem de não exigir contato mecânico para o rastreamento, porém sua aplicação de maneira efetiva requer o conhecimento dos parâmetros e variáveis envolvidas em um processo de rastreamento. Este capítulo introduz o rastreamento óptico, ilustrando-o com um exemplo de rastreamento de um objeto pontual.

2.1. Introdução

Recuperar a posição 3D em uma cena real, a partir de uma ou várias imagens, é um problema inverso, isto é, não é possível resolvê-lo sem a construção de um modelo baseado em dados observados [Chalmond, 2003]. No problema deste trabalho, os dados observados são os pixels, de uma ou várias imagens, capturados com um instrumento, a câmera. A cena real é projetada sobre a geometria em que os pixels estão dispostos. Para a grande maioria das câmeras, esta geometria é uma superfície plana (bidimensional) e com os pixels dispostos em uma grade regular. Os pixels nada mais são que os resultados numéricos da transdução de um dos muitos fotossensores dispostos naquela superfície plana. Transdução é a transformação de um tipo de energia em outra e a energia que se quer transformar é a dos fótons que, depois de interagirem com os objetos da cena real, caem na superfície fotossensorial da câmera. Esta energia dos fótons é, de fato, transformada em energia elétrica, refletindo a quantidade de fótons da cena que colidiram com aquele fotossensor específico. Portanto, de cada fotossensor resultará um sinal que indica, no tempo, a variação da quantidade de luz (quantidade de fótons) detectada por

aquele fotossensor. É preciso, portanto, determinar de quanto em quanto tempo se deve recuperar essa medida, ou seja, para uma fotografia digital (imagem parada) determina-se qual é o tempo de exposição dos fotossensores a uma cena. Para imagens em movimento, este tempo de exposição deve ser suficientemente pequeno para que outra imagem seja obtida, com a sensibilidade desejada.

Esta superfície, geralmente não seria muito útil sem um conjunto óptico associado, isto é, sem um conjunto de lentes. Estas lentes permitem selecionar quais fótons provenientes da cena real devem cair sobre os fotossensores. As lentes, portanto, funcionam como filtros que selecionam qual parte da cena será privilegiada na interação com a câmera.

A Figura 2.1 ilustra a geometria intrínseca de uma câmera. Todos os parâmetros intrínsecos à câmera são aqueles relacionados às características do instrumento. Portanto, somente conhecer tais parâmetros não basta. Se quisermos, resolver o problema proposto por este trabalho, esta (ou estas) câmera deve mergulhar (ou flutuar) em uma cena 3D, isto é, um plano 2D deve mergulhar em um espaço 3D. O posicionamento da câmera no espaço 3D gera um outro conjunto de parâmetros que, não mais, depende do instrumento, mas sim de uma operação que altere sua posição e sua pose, isto é, sua orientação no espaço 3D. Considerando então um espaço euclideano, cujos eixos dimensionais chamaremos de x_w, y_w e z_w, é possível estabelecer, pelo menos, dois sistemas de coordenadas: aquele que toma como referência a câmera e um outro arbitrário.
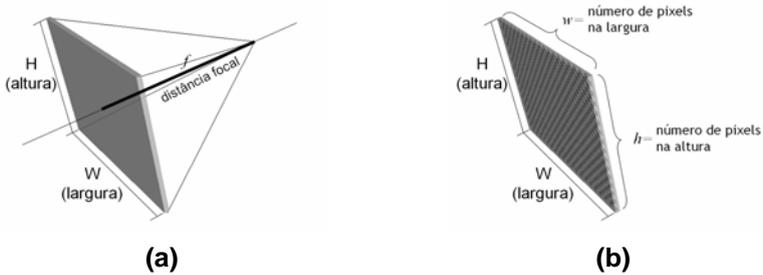
Figura 2.1. Parâmetros intrínsecos de uma câmera. Em (a) aparecem a distância focal e as medidas espaciais da superfície fotossensorial. Em (b) são indicados também o número de pixels em cada dimensão da altura e largura da superfície fotossensorial.

Obviamente está faltando o objeto de interesse nisso tudo, pois é justamente esta a posição que se quer encontrar. Iniciando-se a partir de um objeto pontual, sua posição na cena 3D é representada por uma tripla $P_O=(x_O, y_O, z_O)$. Por outro lado, tudo que se tem é a projeção da imagem deste objeto na câmera, representada por $p_I=(x_I, y_I)$. O problema inverso a ser resolvido é: encontrar P_O conhecendo-se p_I. Claro que isso não seria possível se não houvesse mais informações sobre a geometria do objeto. Da relação entre estes dois pontos (P_O e p_I) só é possível obter a equação da reta ρ que passa pelos pontos P_C (ponto central) e o ponto p_I.
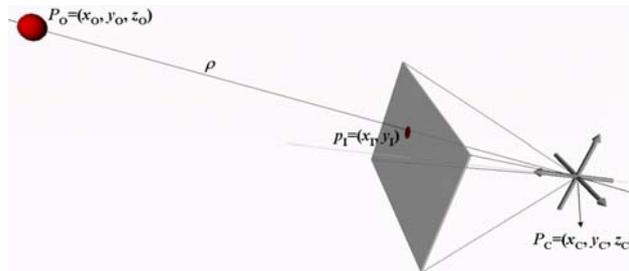
Figura 2.2. Geometria da projeção de um objeto pontual sobre uma única superfície plana.

No entanto, se duas câmeras, C_1 e C_2, forem posicionadas de forma a capturar a projeção do objeto, teremos então duas retas ρ_1 e ρ_2, respectivamente. Encontrar o ponto em que estas duas retas se encontram é resolver o nosso problema.
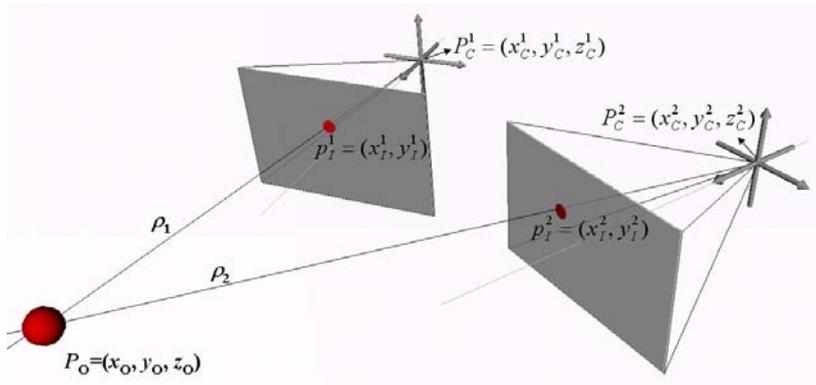
Figura 2.3. Geometria da projeção de um objeto pontual sobre duas superfícies planas.

Ao longo das sessões a seguir será relatado um experimento que pode ser reproduzido com dispositivos e artefatos de baixo custo. Este experimento se baseou na biblioteca de realidade aumentada ARToolkit [Lamb, 2007], seja para fazer a aquisição do vídeo, bem como para visualizar a resolução do problema proposto. A ferramenta para desenvolvimento utilizada foi o MS Visual Studio (versão 6) sobre plataformas PC (Athlon 2600+ com 1GB de memória RAM e Athlon 2400+ com 1GB de memória RAM) com Windows XP Professional. Uma rede local Ethernet de 100 Mbps foi utilizada para conectar os computadores.

2.2. Aquisição e Processamento de Vídeo

O requisito básico para se fazer rastreamento óptico é a aquisição de vídeo em tempo real. As câmeras, ou seja, os dispositivos para se adquirir vídeo em tempo-real, variam muito suas características

e estaria fora do escopo deste capítulo detalhar suas configurações e características. As opções mais simples são as *webcams*, conectadas a um PC por meio de interfaces seriais tais como USB ou IEEE-1394. Neste caso, é preciso estar atento ao tempo de varredura e ao desempenho do próprio PC. Câmeras mais simples, geralmente, possuem drivers reconhecidos pelos sistemas operacionais por concentradores de drivers, tais como o WDM (Windows Device Manager) [Cant, 2005] ou V4L (Video for Linux) [Dirks, 2003]. Há também a opção pelas *camcorders*, que permitem a aquisição de vídeos com maior resolução de pixels e de cores [Jack, 2005]. Porém é preciso reconhecer que um PC nem sempre conseguirá processar vídeos da melhor qualidade, pois isto implicaria em acréscimos, muitas vezes drásticos, da quantidade de memória necessária para se armazenar dados temporários, bem como, de processamento desta grande quantidade de dados.

Se conectarmos duas *webcams* (USB ou IEEE-1394) em um único PC, dificilmente será possível adquirir quadros de cada câmera de maneira satisfatória, pois este mesmo PC teria que conseguir ainda realizar o processamento de cada uma das câmeras. Embora seja possível, os resultados não são adequados. Uma saída seria utilizar placas de aquisição de vídeo para se conectar as câmeras. As interfaces entre as câmeras e estas placas são normalmente de vídeo analógico (S-Video, Vídeo Composto, RGB+sinais de sincronismo) ou então digital (DV, IEEE-1394) [Jack, 2005]. Estas placas são capazes de entregar os quadros de maneira rápida. É preciso saber se o PC terá capacidade de processá-las. Uma outra saída seria estabelecer que cada PC processa uma única câmera. Neste caso, cada PC estaria conectado a uma rede e sua tarefa seria calcular a reta ρ do objeto projetado em sua imagem. Ao final do cálculo da reta, para cada quadro, os coeficientes da reta seriam enviados para um terceiro PC calcular o cruzamento entre as retas. Um problema dessa abordagem é que não há a garantia de que os coeficientes de um par de retas gerados

pelos PCs conectados as suas respectivas câmeras correspondam a quadros gerados no mesmo instante. Por isso, antes, deve ser resolvido o problema de sincronismo: as câmeras devem identificar seus quadros de acordo com o instante da aquisição. Há câmeras de baixo custo que fornecem sinais de sincronismo que podem ser enviados a outras câmeras conectadas [Point Grey, 2006]. Ainda assim, é preciso que dados sobre este sincronismo (por exemplo, um identificador do quadro) sejam propagados juntamente com os coeficientes da reta correspondente.

No experimento deste trabalho não houve a preocupação com o sincronismo, pois um dos objetivos futuros é justamente avaliar o impacto da falta do sincronismo em PCs conectados a uma câmera e ligados a uma rede Ethernet de 100 Mbps. A aquisição de vídeo foi suportada pela biblioteca DirectShow (DSVideoLib) [Pintaric, 2003], utilizada pela versão 2.65 do ARToolkit [Lamb, 2007] e que disponibiliza uma interface chamada `libARvideo`, para controlar a captura dos quadros [Lamb, 2006]. O ARToolkit disponibiliza três funções para controlar o tempo de vida da aplicação: (*i*) inicialização (`init()`); (*ii*) laço principal (`MainLoop()`); e (*iii*) finalização (`cleanup()`). Na primeira, dentre outras eventuais inicializações, a câmera é configurada com a chamada da função `arVideoOpen`, que toma como parâmetro uma cadeia de caracteres. Esta cadeia de caracteres deve abrigar uma sentença cuja sintaxe permite determinar parâmetros intrínsecos da câmera tais como: número de pixels na largura (`videoWidth`) e na altura (`videoHeight`); tipo de câmera e alguns de seus identificadores; a escolha sobre mostrar ou não uma GUI (`showDlg`) quando a câmera estiver sendo iniciada, entre outros. Nas versões mais atuais do ARToolkit (2.7 em diante [Lamb, 2007]), estes parâmetros são configurados em um arquivo XML.

Uma vez configurada a câmera, é recomendável que sejam recuperadas as resoluções (largura e altura) com que realmente a

câmera foi configurada. É possível configurar qualquer dimensão em `arVideoOpen`, porém o hardware da câmera pode não oferecer exatamente a resolução desejada. A função `arVideoInqSize` retorna o número de pixels na largura e altura como parâmetros passados por referência.

Câmeras são dispositivos que, normalmente, dispõem de mecanismos móveis para ajustar a distância focal, ou aproximando, ou distanciando as lentes da superfície sensorial. Além disso, suas características intrínsecas e extrínsecas devem ser reconhecidas por um processo chamado calibração, que será melhor explicado mais adiante, na sessão 2.3.2. Grosso modo, a calibração determina parâmetros da câmera que permitem corrigir distorções, bem como determinar como a câmera estava posicionada, ou ainda, como as lentes estavam ajustadas. A função `arParamLoad` recupera os parâmetros salvos em um processo anterior de calibração. Para isso, é fornecido o caminho e o nome do arquivo contendo estes parâmetros e a variável contendo a estrutura lida desse arquivo. Esta estrutura não abriga todos os dados intrínsecos lidos pela configuração, tais como a largura e altura, que por sua vez, são incluídos em outra variável de calibração juntamente com os parâmetros lidos. Este agrupamento é realizado pela função `arParamChangeSize` e logo após, a função `arInitCparam` carrega estes parâmetros em uma variável alocada em memória, podendo ser exibidos com a função `arParamDisp`. A câmera é finalmente configurada com estes parâmetros, chamando-se a função `argInit`, junto do qual são indicados alguns modos de exibição, tais como a magnificação da imagem a ser exibida na tela, opções de tela cheia, entre outras.

No laço principal, a primeira instrução executada é a de aquisição de um quadro. Esta aquisição ocorre na chamada da função `arVideoGetImage`, que retorna um ponteiro de memória para o local onde a imagem adquirida está armazenada. Esta chamada ocorrerá neste laço, até que a aplicação termine. O

ponteiro de memória não define uma estrutura matricial da imagem. Na verdade, é preciso utilizar a informação das dimensões de largura e altura recuperadas na configuração para reconhecer esta estrutura. Além disso, as informações sobre o número de canais de cores devem ser conhecidas antecipadamente. Geralmente as câmeras permitem que se configure estas características em uma GUI, exibida durante o processo de inicialização. Se a câmera disponibilizar a imagem em 3 canais de cores, com 256 níveis para cada canal, é preciso ter em mente que os dados estarão agrupados em pixels, cada um ocupando 3 bytes. Uma imagem que tivesse 320 pixels na largura e 240 pixels na altura, nesse formato de 3 canais de cores, teria então $320 \times 240 \times 3 = 230.400$ bytes ou 225 Kbytes. É possível também que, além dos canais de cores, haja um canal de transparência, conhecido como canal alfa. Neste caso, a imagem ficaria um pouco maior, ocupando 300 Kbytes. Bem maior que os 75 Kbytes, se a imagem tivesse apenas um canal monocromático. A ordem em que estas informações de cores aparecem também é importante para que se consiga processar a imagem do vídeo. Em formatos de 4 canais, há modos BGRA, para os quais o canal azul (B) é o primeiro byte do pixel, o canal verde (G) é o segundo, o canal vermelho (R) é o terceiro e o canal alfa (A) o quarto. Outro formato comum é o RGB, cuja ordem é a mesma da sigla. Maiores detalhes sobre espaços de cores podem ser encontrados em [Sharma, 2003; Poynton, 1995; Jack, 2005].

Uma vez obtido o ponteiro para a imagem com a função `arVideoGetImage`, o processamento torna-se a tarefa de reconhecer ou identificar os pixels nesta estrutura. Por exemplo, se o objetivo for reconhecer um objeto vermelho na imagem adquirida e, sabendo-se que um quadro de imagem esteja no formato BGRA, tudo o que é necessário fazer é percorrer cada um dos pixels, ou seja, grupos de 4 bytes, e verificar se o valor do canal R é uma vez e meia maior que os canais B e G, por exemplo. Claro que,

dependendo das características do ambiente em que a câmera se encontra, e também da intensidade do vermelho do objeto que se quer reconhecer, ao invés de uma vez e meia, talvez seja melhor usar um fator maior ou menor que 1.5.

Ao longo das discussões a seguir, apenas para efeito de simplicidade, os objetos de interesse serão vermelhos. Para realizar experimentos, essa restrição pode até ser válida e útil. Contudo, para se conseguir um rastreamento mais robusto recorre-se a imagens formadas por fotossensores de radiação fora do espectro visível. A mais comum delas é o infra-vermelho (IR) e a imagem formada por este tipo de radiação é tipicamente monocromática [Dorfmüller, 1999; Sato et al., 2000; Reitinger et al., 2004]. Portanto, para aplicar as abordagens discutidas neste trabalho, troque a detecção de pixels vermelhos pelo processo de limiarização, que funciona como uma detecção de pixels brancos e, todo o restante continuaria válido.

Existem dois grandes grupos de soluções usando IR: (a) um dispositivo que reflete bastante as radiações produzidas por uma fonte; (b) um dispositivo que emite ativamente a radiação. As fontes de IR são LEDs (Diodos Emissores de Luz), e os materiais IR-reflexivos são películas construídas especialmente para esse fim [Dorfmüller, 1999]. As câmeras de IR possuem fotossensores mais sensíveis a esse tipo de radiação. Embora as câmeras convencionais construídas com tecnologia CMOS tenham sensibilidade à radiação IR, elas precisam de filtros para que a luz visível não participe da formação da imagem a ser rastreada [Johnson, 2007].

O Código 2.1 mostra uma função que identifica os pixels de um objeto vermelho em uma cena. Neste código, os parâmetros são: `imPtr`, o ponteiro para a imagem adquirida; `rsPtr`, o ponteiro para um vetor de inteiros com o mesmo número de pixels da imagem, para o qual cada elemento terá um valor 255, se o pixel na posição correspondente for reconhecido como vermelho, ou 0 caso

contrário; `W`, o número de pixels na largura; `H`, o número de pixels na altura.

```
void redTracking(ARUint8 *imPtr, int *rsPtr, int W, int H)
{
  int i, j;
  double mx, my, nv;

  mx = 0; my = 0; nv = 0;

  for (j=0; j < H; j++)
    for (i=0; i < 4*W; i+=4)
    {
         int sr = (int)(imPtr[i+j*4*W+2]);
         int sg = (int)(imPtr[i+j*4*W+1]);
         int sb = (int)(imPtr[i+j*4*W]);
         if ((sr > 1.6*sg) && (sr > 1.6*sb) && (sr > 100))
             rsPtr[i/4+j*W] = 255;
         else
             rsPtr[i/4+j*W] = 0;
    }
}
```

Código 2.1. Código de uma função para identificar pixels vermelhos na memória do quadro adquirida com a biblioteca DSVideoLib.

Perceba que esta função pressupõe que a imagem terá 4 bytes por pixel e também que os bytes estão dispostos na ordem BGRA. O canal A é ignorado e o critério *ad hoc* para reconhecer um pixel como vermelho é que o valor do canal R deve ser maior que 100 (cada canal pode assumir valores entre 0 e 255) e este mesmo valor deve ser 1.6 vezes maior que os canais G e B. O que o Código 2.1 realiza é a classificação dos pixels de uma imagem como vermelhos ou não. No entanto, para reconhecer um objeto da cena, que está sendo projetada nesta imagem, é conveniente determinar uma representação para este objeto.

Se o objetivo for reconhecer um objeto pontual vermelho, a representação mais conveniente é um ponto único, ou seja,

buscamos aqui a posição de um pixel deste objeto. Mas, qual? A função do Código 2.1, muito provavelmente, classificará muitos pixels desta imagem como vermelhos e, portanto, como pertencentes ao objeto de interesse. Uma solução para este problema é segmentar a imagem e usar representações que propiciem a distinção entre objetos de interesse e outros que não sejam relevantes para a aplicação. Para isso lança-se mão de técnicas de análise de formas para caracterizar os objetos [Costa et. al., 2000].

2.3. Modelos de Projeção

Modelo é uma abstração, ou seja, uma idealização da realidade. A realidade de uma câmera, por exemplo, é que o dispositivo fabricado pode distorcer a cena projetada, tanto na disposição dos pixels distribuídos espacialmente na imagem, quanto das cores que podem não corresponder ao convencionado.

Nesta seção será apresentado o modelo de projeção em perspectiva. Este é um modelo simples e corresponde a uma simplificação da geometria da câmera, chamado modelo "*pin hole*", por não considerar as características das lentes [Ma et alli, 2004]. Retomando a Figura 2.2, o objetivo é encontrar a equação de uma reta que passe pelos pontos p_I e P_O. Para isso, é preciso relacioná-los.
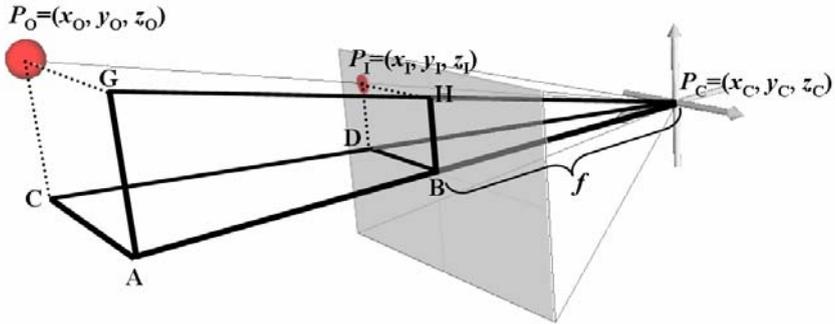
Figura 2.4. Modelo de projeção em perspectiva mostrando a geometria da formação da imagem e os 3 pontos de interesse deste trabalho: P_O, as coordenadas do objeto pontual na cena; P_I, as coordenadas da projeção do objeto na imagem; e P_C, o centro de projeção da câmera.

Na Figura 2.4 está ilustrada a geometria deste modelo onde, de um lado estão as coordenadas referentes ao objeto pontual P_O e de outro as coordenadas do centro de projeção da câmera P_C. Entre estes dois pontos, mas na mesma reta, está o ponto projetado na imagem P_I. É fácil perceber que há um conjunto de triângulos semelhantes. Em particular, os triângulos AP_CG e BP_CH são semelhantes, assim como os triângulos AP_CC e BP_CD. Portanto é direto estabelecer a relação entre os tamanhos de segmentos de retas de acordo com (1).

Se P_C for adotado como a origem deste espaço de coordenadas, ou seja, se P_C =(0, 0, 0), obtem-se um relacionamento entre estas coordenadas, no qual o valor de f determina a proporcionalidade entre a posição desse objeto (inclusive seu próprio tamanho) e a posição de sua projeção na imagem. Em outras palavras, f é um fator de escala entre o objeto e sua imagem. Desenvolvendo algebricamente estas proporções triangulares e reconhecendo nos segmentos de reta destes triângulos as

respectivas coordenadas de um sistema cuja origem é P_C, então o relacionamento pode ser escrito como (1).

$$\frac{\overline{BH}}{\overline{AG}} = \frac{\overline{BD}}{\overline{AC}} = \frac{\overline{BP_C}}{\overline{AP_C}} \Leftrightarrow \frac{x_I}{x_O} = \frac{y_I}{y_O} = \frac{f}{z_O} \qquad (1)$$

Destas proporções resultam as duas equações em (2).

$$\frac{x_I}{f} = \frac{x_O}{z_O} \text{ e } \frac{y_I}{f} = \frac{y_O}{z_O} \qquad (2)$$

Este modelo ficaria mais sintético se fosse representado na forma matricial (3).

$$\begin{bmatrix} x_I \\ y_I \end{bmatrix}_C = \frac{f}{z_O} \begin{bmatrix} x_O \\ y_O \end{bmatrix}_C \qquad (3)$$

A representação do modelo de projeção por coordenadas homogêneas é particularmente útil, pois neste sistema de coordenadas um ponto em um plano é representado como uma linha no espaço. Qualquer representação em coordenadas cartesianas, para ser transformada em coordenadas homogêneas, deve ter uma dimensão a mais que escale as dimensões originais. No exemplo em (4), um ponto como coordenada cartesiana em duas dimensões é transformado em um ponto como coordenada homogênea, com as dimensões originais acrescidas de um fator de escala *w* [Schalkoff, 1989].

$$\begin{bmatrix} x \\ y \end{bmatrix}_C \Leftrightarrow \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix}_H = \begin{bmatrix} x \\ y \\ w \end{bmatrix}_H \qquad (4)$$

Analogamente, para um ponto de três dimensões em coordenadas cartesianas, o exemplo (5) mostra a transformação para coordenadas homogêneas usando o fator de escala *w*.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_C \Leftrightarrow \begin{bmatrix} x/w \\ y/w \\ z/w \\ 1 \end{bmatrix}_H = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}_H \qquad (5)$$

Perceba que as coordenadas com fator de escala normalizado (w=1) ou não, representam o mesmo ponto. Voltando ao relacionamento entre p_I e P_O, observe que (3) poderia ser reescrito em coordenadas homogêneas de acordo com (6).

$$z_O \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix}_H = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_O \\ y_O \\ z_O \\ 1 \end{bmatrix}_H \Leftrightarrow z_O p_I = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P_O \qquad (6)$$

A profundidade do objeto (z_O) que multiplica p_I pode ser interpretada como um fator de escala, ou seja, uma constante que pudesse ser arbitrariamente definida, pois o que é possível se saber sobre a posição onde está o objeto de interesse é que ele reside sobre a linha ρ. O modelo acima pode ainda desmembrar a matriz de projeção (7), isolando o modelo de projeção, representado pela matriz Π_O do parâmetro intrínseco envolvido, que por enquanto é apenas a distância focal f e é representado pela matriz K_f.

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_O} \qquad (7)$$

Tudo o que é preciso para calcular a equação de uma reta é um par de pontos. No nosso caso, teremos que conhecer P_C e P_I. Se P_C é a origem adotada, então P_C=(0, 0, 0). Por sua vez, P_I é resultante da identificação do rastreamento da projeção do objeto sobre a imagem (veja Seção 2.2), acrescida da coordenada z_I=f. As

coordenadas de P_I são facilmente reconhecidas no plano da imagem, contudo é preciso transformar esta posição de pixels em coordenadas de um sistema que deve incluir o objeto. Além disso, o modelo deve incorporar as características da lente da câmera. As retas ρ_1 e ρ_2 da Figura 2.5 devem residir em um único espaço de coordenadas, pois é no cruzamento entre elas que estará o objeto. Portanto, é necessário encontrar uma referência única para as duas câmeras. É sobre isso que trataremos na próxima seção.
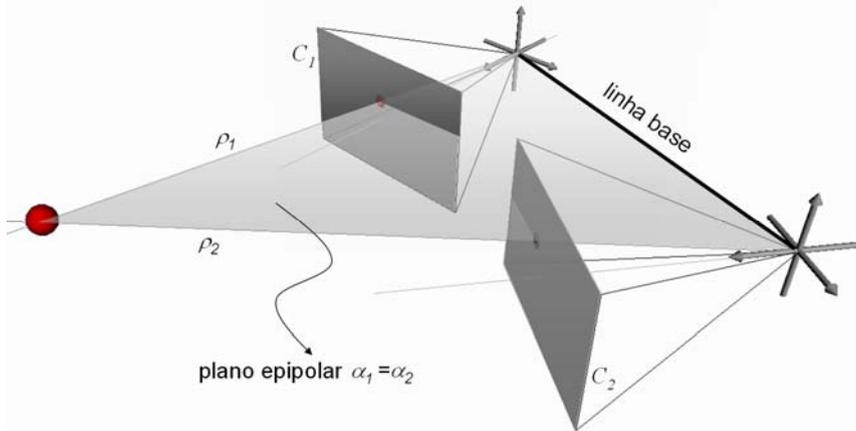
Figura 2.5. Modelo de projeção binocular que obedece a restrição epipolar. Observe que o plano α_1, contendo a linha base e a reta ρ_1, e o plano α_2, contendo a linha base e a reta ρ_2, são coincidentes.

2.3.1. A Restrição Epipolar

Se a câmera estiver apontada para o objeto de interesse, ou melhor, se o objeto de interesse estiver dentro do campo de visão da câmera, então é possível calcular uma reta que passe pelo centro de projeção (P_C) e pela projeção do objeto na imagem (P_I), de acordo com a referência da câmera (origem em P_C). O próximo passo é colocar outra câmera, cujo campo de visão inclua o objeto de

interesse. Porém ainda falta: 1) estabelecer a restrição epipolar e também 2) encontrar uma referência única.

A linha que liga os dois centros de projeção das duas câmeras é chamada linha-base. Se duas retas residem em um plano, então é fácil perceber que a reta passando pela linha-base e aquela passando pelo centro de projeção (P_C) e pelo objeto (P_O) residem em um plano, chamado plano epipolar (veja Figura 2.5).

A restrição epipolar estabelece que os planos de cada câmera devem ser coincidentes, ou de outra forma, as três retas: a que passa por P_{C1} e P_{O1}, a que passa por P_{C2} e P_{O2} e a que passa pela linha-base devem pertencer a um mesmo plano. No final das contas, o que esta restrição impõe é que as linhas dos objetos se cruzem. Veja na Figura 2.5, que os eixos das câmeras devem também residir em um mesmo plano.
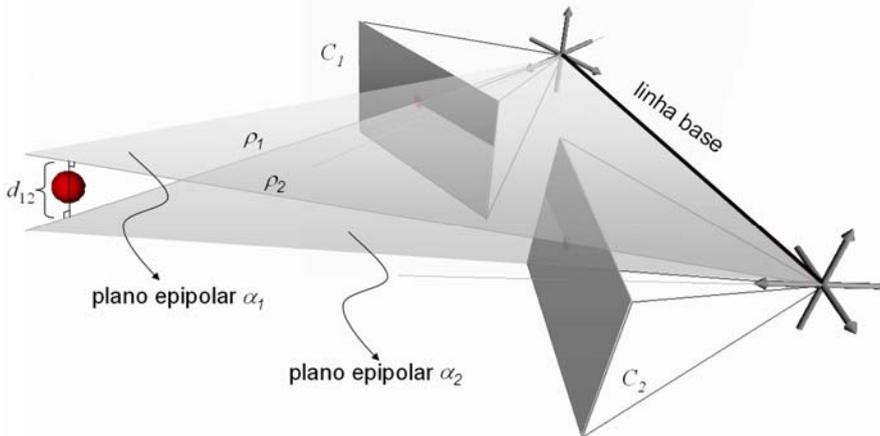
Figura 2.6. Modelo de projeção binocular que não obedece a restrição epipolar. Observe que o plano α_1, contendo a linha base e a reta ρ_1, e o plano α_2, contendo a linha base e a reta ρ_2, são distintos.

Para garantir essa restrição, ou pelo menos minimizá-la, é preciso criar um mecanismo que possa medir a distância d_{12} entre as duas retas (ρ_1 e ρ_2) dos objetos de cada câmera, ou então construir uma montagem suficientemente precisa e acurada, que permita localizar a câmera em ângulos e posições fixas e mensuráveis (veja a Figura 2.6).

A alternativa proposta neste trabalho é usar, como referência comum, um marcador. Embora o ARToolkit recupere a posição do marcador com pouca precisão, este exercício permitirá que se prescinda de uma estrutura montada em uma trave ou mecanismos que assegurem o posicionamento e a pose das câmeras.

O sistema de coordenadas do marcador é obtido com o rastreamento desse marcador, do qual é identificado, na imagem, aquele objeto mais similar a um quadrado e então, seus quatro cantos são extraídos. Cada um destes quatro cantos são origens de 3 vetores nas direções dos outros 3 cantos. Esta abordagem é exposta com mais detalhes em [Kato et. al., 1999]. No ARToolkit a matriz que transforma o sistema de coordenadas, cuja origem é o centro de projeção da câmera, no sistema de coordenadas com origem no centro do marcador é calculada pela função `arGetTransMat`. Para resolver o problema deste trabalho é preciso, para cada câmera, encontrar a representação dos pontos correspondentes ao centro de projeção (P_{C1} e P_{C2}) e à projeção do objeto na imagem (P_{I1} e P_{I2}). Desta maneira, residindo em um sistema de coordenadas único estão as duas retas cujo cruzamento seria a posição procurada.
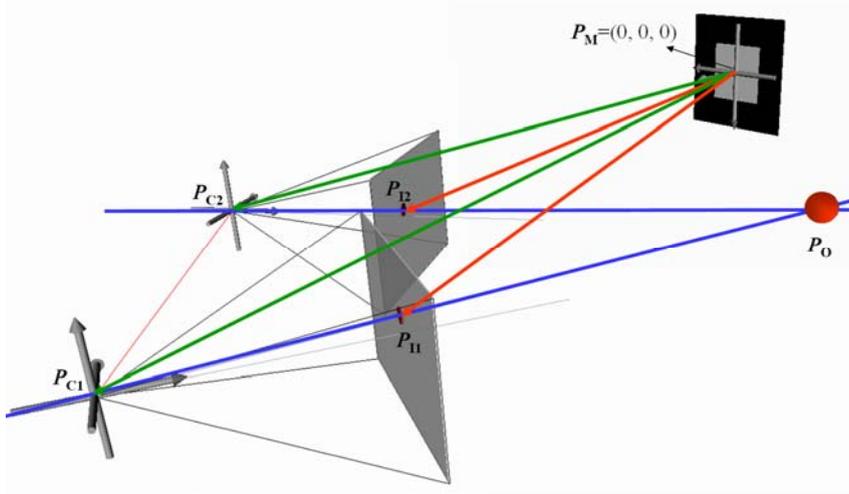
Figura 2.7. Modelo de projeção binocular com um ponto de referência comum calculado a partir de um marcador (P_M).

Na verdade, não é tão simples assim, pois se estivéssemos procurando este cruzamento, seria pouco provável encontrá-lo, pois dificilmente conseguiríamos um posicionamento de câmeras que obedecesse à restrição epipolar. Por isso, o que é possível é calcular a distância d_{12} (veja Figura 2.6) entre as duas retas. Esta distância será tanto melhor quanto menor e, ainda pode ser utilizada como uma indicação de qualidade do rastreamento.

Para calcular d_{12}, é preciso recorrer a dois pontos da reta ρ_1 e dois de ρ_2, representados aqui como vetores. Um dos vetores em ρ_1 pode ser P_{C1} ($\vec{p}_{11}$) e o outro P_{I1} ($\vec{p}_{12}$). Analogamente para ρ_2, P_{C2} ($\vec{p}_{21}$) e P_{I2} ($\vec{p}_{22}$) podem ser os vetores utilizados [Goodman et. al., 2004].
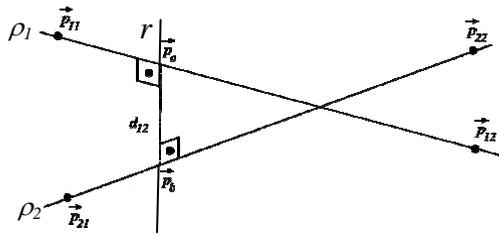
Figura 2.8. Distância entre duas retas que não se cruzam.

Inicialmente representa-se as retas ρ_1 e ρ_2 conforme uma reta no espaço tridimensional no qual residem $\vec{p}_{11}$, $\vec{p}_{12}$ e um ponto $\vec{p}_a$ qualquer de ρ_1 e $\vec{p}_{21}$, $\vec{p}_{22}$ e um ponto $\vec{p}_b$ qualquer de ρ_2. Estas retas também são parametrizadas por um coeficiente de inclinação m_a e m_b, respectivamente para ρ_1 e ρ_2. As equações para as duas retas podem ser visualizadas em (8).

$$\begin{aligned} \rho_1 &: \vec{p}_a = \vec{p}_{11} + m_a(\vec{p}_{12} - \vec{p}_{11}) \\ \rho_2 &: \vec{p}_b = \vec{p}_{21} + m_b(\vec{p}_{22} - \vec{p}_{21}) \end{aligned} \tag{8}$$

A ilustração da Figura 2.8 mostra que estas retas não necessariamente se cruzam e, portanto, calcular d_{12} é, de fato, calcular a distância entre elas. A distância entre duas retas é definida pelos pontos de cada uma destas retas, pelos quais seja possível construir um segmento de reta perpendicular a ambas, denotada como *r*.

$$\begin{aligned} (\vec{p}_a - \vec{p}_b)\cdot(\vec{p}_{12} - \vec{p}_{11}) = 0 \\ (\vec{p}_a - \vec{p}_b)\cdot(\vec{p}_{22} - \vec{p}_{21}) = 0 \end{aligned} \tag{9}$$

Na representação apresentada, $\vec{p}_a$ e $\vec{p}_b$ são os dois vetores que, agora, poderiam ser impostos como representantes destes dois pontos. Portanto, tomando-se os vetores resultantes das diferenças entre $\vec{p}_a$ e $\vec{p}_b$ ($\vec{p}_a - \vec{p}_b$) em *r*, entre $\vec{p}_{12}$ e $\vec{p}_{11}$ em ρ_1 ($\vec{p}_{12} - \vec{p}_{11}$) e entre $\vec{p}_{22}$ e $\vec{p}_{21}$ em ρ_2 ($\vec{p}_{22} - \vec{p}_{21}$), impõe-se as restrições da perpendicularidade entre as retas ρ_1 e *r*; e entre ρ_2 e *r*. Desta

maneira, em (9) os produtos escalares entre os vetores diferença que direcionam r e ρ_1 devem ser nulos, assim como os que direcionam r e ρ_2.

$$\vec{v}_{21}^{11} = (p_{11} - p_{21}) \qquad \vec{v}_{21}^{22} = (p_{22} - p_{21}) \qquad \vec{v}_{11}^{12} = (p_{12} - p_{11}) \tag{10}$$

Manipulando algebricamente a expressão resultante da substituição de (8) em (9) e assumindo os três vetores obtidos pela diferença entre os pontos das retas ρ_1 e ρ_2 em (10), é possível chegar ao valor de m_a, conforme (11).

$$m_a = \frac{(\vec{v}_{21}^{11} \cdot \vec{v}_{21}^{22})(\vec{v}_{21}^{22} \cdot \vec{v}_{11}^{12}) - (\vec{v}_{21}^{11} \cdot \vec{v}_{11}^{12})(\vec{v}_{21}^{22} \cdot \vec{v}_{21}^{22})}{(\vec{v}_{11}^{12} \cdot \vec{v}_{11}^{12})(\vec{v}_{21}^{22} \cdot \vec{v}_{21}^{22}) - (\vec{v}_{21}^{22} \cdot \vec{v}_{11}^{12})(\vec{v}_{21}^{22} \cdot \vec{v}_{11}^{12})} \tag{11}$$

O valor de m_b é obtido a partir do valor de m_a e também dos vetores diferença em (10), como pode ser visto em (12).

$$m_b = \frac{(\vec{v}_{21}^{11} \cdot \vec{v}_{21}^{22}) + m_a(\vec{v}_{11}^{12} \cdot \vec{v}_{21}^{22})}{(\vec{v}_{21}^{22} \cdot \vec{v}_{21}^{22})} \tag{12}$$

Uma vez obtidos m_a e m_b, para encontrar $\vec{p}_a$ e $\vec{p}_b$ basta apenas substituí-los em (8). O valor de d_{12} nada mais é que o tamanho (ou módulo) do vetor $\vec{p}_a - \vec{p}_b$, como está mostrado em (13).

$$d_{12} = \left\| \vec{p}_a - \vec{p}_b \right\|_2 = \sqrt{(\vec{p}_a - \vec{p}_b) \cdot (\vec{p}_a - \vec{p}_b)} \tag{13}$$

Ao final, as coordenadas do objeto pontual rastreado podem ser escolhidas como sendo $\vec{p}_a$ ou $\vec{p}_b$, ou ainda o ponto médio.

2.3.2. Considerações sobre Calibração de Câmeras

Câmeras são dispositivos totalmente dependentes de instrumentos ópticos que, por suas características de fabricação, ou por sua interação com o operador, ou ainda pelas condições ambientais, podem distorcer a projeção sobre a superfície fotossensorial. Na tentativa de corrigir estas distorções, a formação da imagem sobre

as câmeras é idealizada considerando algumas características que sejam conhecidas e que possam reduzir o erro de suas projeções.

O modelo de projeção em perspectiva de uma câmera "*pin hole*" é uma destas idealizações (veja Seção 2.3), mas como o próprio nome diz, não considera o uso das lentes. Veja bem que em (7) a matriz de projeção foi desmembrada em duas: K_f, a matriz de calibração e Π_O, a matriz de projeção. K_f considerava, naquela seção, apenas a distância focal e, portanto, um objeto mais distante apareceria f vezes menor, quando projetada sobre a superfície fotossensorial. Outras características, no entanto, podem ser consideradas, tais como o tamanho do pixel, a distorção radial ou qualquer outro parâmetro que caracterize a câmera. Claro que, quanto mais parâmetros forem considerados, mais complexo é o modelo e mais difícil será também calibrar a câmera. Dependendo do parâmetro, nem sempre uma matriz K_f 3×3 será suficiente para comportá-los.

Para o propósito deste trabalho, o processo de calibração de câmeras resume-se em encontrar os parâmetros da câmera e dispô-los na forma matricial K_f 3×3, para que se corrija a posição p_I, ainda no sistema de coordenadas cuja referência é o centro de projeção da câmera.

Existem muitos processos de calibração de câmeras, uns muito simples outros nem tanto [Silva, 2003]. Tudo depende dos requisitos de qualidade para a resposta. Há aplicações que são críticas quanto a precisão, outras nem tanto. O ARToolkit disponibiliza um processo de calibração que salva os parâmetros em um arquivo. Este processo se encontra descrito em [Consularo et al, 2004]. É um processo simples, porém trabalhoso e sem possibilidade de contorná-lo. Uma vez com a câmera calibrada, é possível confiar melhor nas coordenadas de p_I obtidas, porém sempre tendo em mente que a câmera é um instrumento e todo e qualquer instrumento apresenta incertezas.

2.4. Usando o ARToolkit para visualizar o rastreamento

Modelo de projeção em perspectiva, calibração de parâmetros intrínsecos e extrínsecos, tudo isso já faz parte da implementação do ARToolkit e está disponível na forma de bibliotecas.

A biblioteca ARToolkit para Realidade Aumentada, `ARlib`, implementa o modelo de projeção e a própria calibração, pois estas tarefas são exigidas para a detecção da posição e da pose de um marcador. O marcador é um quadrado preto em fundo branco, que é reconhecido na chamada da função `arDetectMarker`, cujos parâmetros de entrada são: um ponteiro para os pixels da imagem de entrada e um limiar para distinguir o fundo de objetos. A saída ocorre passando-se duas referências: um ponteiro para uma estrutura, que conterá dados sobre o marcador reconhecido, e um outro ponteiro para um valor inteiro que receberá o número que identifica o marcador reconhecido [Lamb, 2006].

A função `MainLoop`, no exemplo *exview* disponível nas distribuições do ARToolkit, abriga uma seqüência de tarefas para cada quadro em processamento: 1) os dados da imagem do quadro que foi capturado são apontados por um ponteiro obtido pela chamada da função `arVideoGetImage`; 2) deve ser realizado o processamento da imagem deste quadro, tal como aquele do Código 2.1; 3) se este processamento implicar em alteração desta imagem, esta alteração deve ocorrer logo após o processamento; 4) configurações sobre o modo como o OpenGL será utilizado; 5) a imagem aumentada com os objetos 3D é exibida chamando-se a função `argDispImage`; 6) um marcador é buscado na imagem com a chamada de `arDetectMarker`; 7) se o marcador foi detectado, então suas posição e pose são recuperadas com a chamada da função `arGetTransMat`; 8) uma cena é desenhada com o OpenGL, cena esta contendo um objeto sobre o marcador representando os 3 eixos ortogonais de seu espaço, uma câmera na posição e na pose

relativa ao marcador detectado e uma superfície plana, coplanar à superfície do marcador.

Com o propósito de ilustrar este trabalho, o exemplo do *exview* foi desmembrado em duas aplicações: uma servindo apenas como rastreador (*ARTracker*) e outra servindo como um visualizador 3D da geometria do rastreamento (*ARViewer*). O *ARTracker* funciona como um servidor em uma arquitetura cliente-servidor (veja Figura 2.9), que pode ser requisitado para enviar dois tipos de informação, *i*) os parâmetros de câmera: a matriz de calibração 3×3 (K_f), a matriz de projeção 3×4 ($\mathbf{M}_T$) e os fatores de distorção 1×4 (D); *ii*) as coordenadas atualizadas da projeção do objeto sobre a imagem (x_I, y_I). O serviço é realizado por uma *thread* iniciada quando a aplicação é também iniciada. Na linha de comando, se nenhum número de porta for informado, assume-se o identificador 1200. Se dois computadores estiverem, cada um com uma câmera conectada, o *ARTracker* deve ser executado em cada um deles.

$$K_f = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (14)$$

A matriz de calibração é formada por parâmetros dispostos de acordo com (14), onde s_x e s_y são os fatores de magnificação da imagem nas direções x e y, respectivamente e o_x e o_y são as coordenadas do centro de projeção na imagem.

A matriz de projeção $\mathbf{M}_T$ é o operador de rotação e translação que, transforma o sistema de coordenadas da câmera para o sistema de coordenadas cuja origem é o centro do marcador. Ela é obtida com a chamada da função `arGetTransMat`, que toma como entrada as informações sobre o marcador (tamanho, centro, identificador, entre outras). O fator de distorção D é um vetor

formado por 4 valores: as coordenadas c_x e c_y do centro de projeção da imagem, a distância focal (*f*) e a taxa de aspectos do pixel.

O *ARViewer* (veja Figura 2.9) pode executar em um terceiro computador, ou ainda como uma tarefa de um dos dois computadores executando o *ARTracker*. Convém salientar que, se a conexão de duas câmeras em um mesmo computador for possível, os três processos, ou seja, os dois *ARTrackers* para as câmeras esquerda e direita, mais o *ARViewer* podem operar em um único computador, tomando-se o cuidado de adotar identificadores diferentes para a porta de cada *ARTracker*. O *ARViewer* é um cliente de dois computadores ligados, cada um a uma câmera que requer, na sua chamada, os endereços IP de cada um dos computadores executando os *ARTrackers*, acompanhados de suas respectivas portas. Novamente, duas *threads* executam estas requisições continuamente. Nestas *threads* podem ser requisitados continuamente os parâmetros de câmera e as coordenadas do ponto rastreado na imagem, ou então apenas as coordenadas do ponto rastreado, se os parâmetros da câmera já foram anteriormente requisitados. Uma vez com os dados requisitados, as duas câmeras são desenhadas em uma cena OpenGL que já contém um plano de referência do marcador, representada por uma superfície plana quadriculada e um eixo ortogonal sobre a origem. As câmeras são visualizadas em conjunto, com linhas que indicam: o seu eixo óptico, o *frustrum*, e uma linha passando pelo centro de projeção da câmera e pela coordenada da projeção do objeto pontual na imagem (veja Figura 2.1*a*). Estas linhas são desenhadas conforme recebem os dados de cada requisição, ficando limitada ao tráfego da rede e também de suas características.
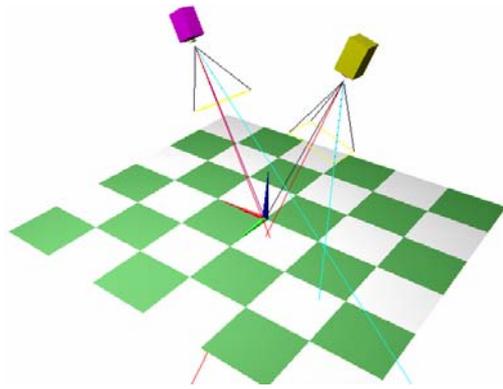
Figura 2.9. Execução do ARTracker mostrando as duas câmeras conectadas, cada uma a um computador (coluna da esquerda). Execução do ARViewer (coluna da direita) mostrando o eixo ortogonal sobre o plano quadriculado que representa o eixo de coordenadas de referência do marcador. Os *frustrums* de cada câmera também são exibidos, bem como as retas do eixo óptico de cada câmera, ρ_1 e ρ_2.

Depois de receber os dados de cada instância do *ARTracker*, ou seja, de posse dos parâmetros da câmera e das coordenadas p_I da projeção do objeto sobre a imagem, o problema passa a ser apenas encontrar os pontos P_C e P_I de cada câmera, de acordo com o sistema de coordenadas do marcador. Para se obter o P_C basta recuperar a última coluna da matriz de projeção $\mathbf{M}_T$. Note que existe uma matriz de projeção para cada câmera. A informação que se tem sobre os pontos p_I, no entanto, ainda residem no plano da imagem e, portanto, são bidimensionais com coordenadas (x_I, y_I). É preciso, antes de tudo corrigir sua distorção aplicando-se a matriz de calibração, ou seja, calcular (x_I', y_I') de acordo com (15).

$$x'_I = f\,\frac{(x_I - c_x)}{\alpha_x} \qquad y'_I = f\,\frac{(y_I - c_y)}{\alpha_y} \tag{15}$$

Uma vez corrigida a distorção, deslocadas as coordenadas do centro, a coordenada z'_I deve ser determinada como *f*, pois no sistema de coordenadas da câmera é a essa distância que o plano da imagem se encontra do ponto P_C. Portanto, agora as coordenadas do ponto projetado na imagem são (x'_I, y'_I, f). A matriz de projeção $\mathbf{M}_T$ é, na verdade, o operador que transforma o sistema de coordenadas da câmera no sistema de coordenadas do marcador. No entanto, o que queremos é justamente o inverso, ou seja, obter o ponto P_I de acordo com o sistema de coordenadas do marcador. Portanto, invertendo-se $\mathbf{M}_T$ e aplicando este resultado à coordenada homogênea de P_I (ainda em coordenadas da câmera) obtemos suas coordenadas tendo como referência o marcador, conforme (16).

$$P_I = \mathbf{M}_T^{-1} P'_I \Rightarrow \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} x'_I \\ y'_I \\ f \\ 1 \end{bmatrix} \tag{16}$$

A inversão de uma matriz 3x4 não é única e também não é um processo trivial. O ARToolkit disponibiliza uma função `arUtilMatInv`, cuja entrada é uma matriz 3x4 e sua saída é outra matriz 3x4, ambas como parâmetros por referência. Com P_C e P_I calculados para cada câmera, encontra-se o cruzamento entre as duas retas de cada câmera, como explicado na Seção 2.3.1.

2.5. Considerações Finais

Neste trabalho, foram apresentados didaticamente, os passos necessários para o rastreamento de pontos em um espaço real por um arranjo binocular de câmeras, tendo como referência um marcador detectado pelo ARToolkit.

O trabalho proposto permite estabelecer uma métrica de desempenho, que pode ser realimentada para minimizar a distância d_{12}, na tentativa de satisfazer a restrição epipolar.

A construção de dispositivos de rastreamento para realidade virtual e aumentada usando câmeras pode partir das técnicas simples explicadas neste trabalho. Se, por exemplo, soluções algorítmicas do problema da correspondência entre pontos das duas imagens, forem adicionadas, é possível construir dispositivos compostos, possibilitando um rastreamento mais rico.

Câmeras menos limitadas podem ser utilizadas, muito provavelmente tornando mais precisas e rápidas as respostas. O uso de redes de alta velocidade para a comunicação das posições e parâmetros entre os computadores permitiria garantir dispositivos mais confiáveis. A garantia, por exemplo, oferecida pelo sincronismo entre as câmeras é fator determinante para uma maior confiabilidade.

O espaço de interação do dispositivo é limitado pela intersecção dos *frustrums* das câmeras. Um dispositivo que permitisse a detecção do volume dessa intersecção entre *frustrums* e, posteriormente, realimentasse dispositivos de movimentação da câmera (vergência, proximidade e inclinação) poderia resolver este problema, na busca de mais robustez da resposta.

Câmeras com campos de visão mais amplos poderiam ser experimentadas para contrapor a construção de um dispositivo para movimentar as câmeras, conforme a solução do parágrafo anterior.

Como se percebe, este tipo de trabalho é ponto de partida para a construção de dispositivos de interação baseados em câmera, com aplicações que tipicamente exijam pouca precisão na interação, como é o caso de interfaces com o usuário.

2.7. Referências Bibliográficas

Cant, C. (2005) "Writing Windows WDM Device Drivers", Academic Press, 580pp.

Chalmond, B. (2003) "Modeling and Inverse Problems in Image Analysis", Springer, 328pp.

Consularo, L. A.; Coelho, R. C.; Fernandes, M. M. (2004) "Tutorial sobre o ARToolKit", http://www.consularo.com/artoolkit, (visitada em 04/03/2007).

Costa, L. F.; Cesar Jr., R. M. (2000) "Shape Analysis and Classification. Theory and Practice", CRC Press, 659pp.

Dirks, R. (2003) "Video for Linux Two", http://www.thedirks .org/v4l2/, (visitada em 04/03/2007).

Dorfmüller, K. (1999) "Robust Tracking for Augmented Reality using Retroreflective Markers", *Computers and Graphics*, v23, p795-800.

Goodman, J. E.; O'Rourke, J. (2004) "Handbook of Discrete and Computational Geometry", 2ª ed., Chapman and Hall/CRC Press, 1512pp.

Jack, K. (2005) "Video Demystified. A Handbook for Digital Engineer", Newnes, 927pp.

Johnson, G. (2007) "How to make a webcam work in infrared", http://www.hoagieshouse.com/IR/, (visitada em 04/03/2007).

Kato, H.; B., Mark (1999) "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, p85-94.

Lamb, P. (2006) "ARToolKit Documentation", http://artoolkit. sourceforge.net/apidoc/, (visitada em 04/03/2007).

Lamb, P. (2007) “ARToolKit Developer Page”, http://sourceforge.net/projects/artoolkit, (visitada em 04/03/2007).

Lamb, P. (2007) “ARToolKit Home Page”, http://www.hitl.washington.edu/artoolkit, (visitada em 04/03/2007).

Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S. (2004) “An Invitation to 3D Vision. From Images to Geometric Models”, Springer, 526pp.

Pintaric, T. (2003) “DSVideoLib”, http://sourceforge.net/projects/dsvideolib, (visitada em 04/03/2007)

Point Grey Research (2006) “Dragonfly Express”, http://www.ptgrey.com/products/dx, (visitada em 04/03/2007).

Poynton, C. (1995) “A Guided Tour of Color Space”, Proceedings of the SMPTE Advanced Television and Electronic Imaging Conference, p165-170.

Reitinger, B.; Bornik, A.; Beichel, R.; Werkgartner, G.; Sorantin, E. (2004) “Augmented Reality based Measurement Tools for Liver Surgery Planning”, Bildverarbeitung für die Medizin (BVM), Springer, p274-278.

Sato, Y.; Kobayashi, Y.; Koike, H. (2000) “Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface”, Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition, p462-467.

Schalkoff, R. J. (1989) “Digital Image Processing and Computer Vision”, John Wiley and Sons, 489pp.

Sharma, G. (2003) “Digital Color Imaging Handbook”, CRC Press, 764pp.

Silva, L. C. (2003) “Método Robusto para a Calibração de Câmeras em Estereofotogrametria”, Tese de Doutorado – COPPE – UFRJ, 117pp.

Capítulo 3

Técnicas de Interação para Ambientes de Realidade Virtual e Aumentada

Judith Kelner e Veronica Teichrieb

Grupo de Pesquisa em Realidade Virtual e Multimídia, Centro de Informática – Universidade Federal de Pernambuco (GRVM CIn UFPE)
Av. Prof. Moraes Rego, S/N – Prédio da Positiva – 1º Andar – Cidade Universitária – 50670-901 – Recife – PE – Brasil

{jk,vt}@cin.ufpe.br

Abstract

Software usability is highly dependent on the supported interaction at the interface level. A number of techniques have been described in the literature, and some of those used in the context of virtual and augmented realities together with implementation aspects are presented in this chapter.

Resumo

A usabilidade de um software é dependente das técnicas de interação disponíveis na sua interface. Várias dessas técnicas encontram-se descritas na literatura, e algumas usadas nos ambientes de Realidade Virtual e Aumentada, com uma breve descrição sobre sua implementação, são descritas neste capítulo.

3.1. Introdução

A interação entre pessoas existe desde o surgimento das primeiras civilizações humanas. Diferentes culturas utilizaram diversas formas para a troca de informações, numa evolução contínua que teve início com os primeiros desenhos rupestres nas cavernas, passando pelos sinais de fumaça, os sons dos tambores, as pinturas nas telas, a imprensa escrita e falada, o telefone, a televisão e, finalmente, o computador e toda parafernália tecnológica existentes atualmente.

No contexto de interface homem-máquina, interação é a maneira com que o usuário se comunica com a aplicação, podendo esta comunicação ocorrer através de dispositivos ou de forma simbólica [Schneiderman e Plaisant, 2004].

De acordo com Bowman, "interação é um método que permite a um usuário realizar uma tarefa através da interface do usuário. Uma técnica de interação inclui tanto componentes de *hardware* (dispositivos de entrada/saída) quanto de *software*. As técnicas de interação utilizadas nos componentes de *software* são responsáveis por mapear a informação de um dispositivo de entrada em alguma ação dentro do sistema, e por mapear a saída do sistema de forma que esta possa ser interpretada pelos dispositivos de saída" [Bowman et al., 2004].

Existe um dito popular, que não se deve adquirir um livro apenas pela aparência da sua capa. Isto pode ser válido no contexto dos livros, mas considerando-se *software*, a experiência indica que vários usuários deixam de usar um *software* sofisticado por causa de uma interface mal projetada e das dificuldades de interagir através da mesma.

No início da era computacional, não se deu muita importância ao processo de interação homem-máquina (HCI). A prioridade era obter um processamento preciso dos dados. Com a evolução e disseminação dos computadores pessoais cresceu a

necessidade de serem adotadas metodologias específicas para a HCI. Os usuários não estão dispostos a consumir seu tempo precioso com aplicações que possuam interfaces que utilizam técnicas primárias de interação. A busca por estratégias avançadas de interação contempla, atualmente, grande parte do tempo dos projetistas quando se inicia o processo de desenvolvimento de um novo *software* ou de um novo dispositivo.

Inúmeros são os benefícios que um bom projeto de interação pode agregar a um *software*. Entre estes, podem-se citar a usabilidade do sistema, menor curva de aprendizagem, localização e uso de todas as potencialidades da aplicação, otimização do tempo do usuário na busca pela informação, entre outros.

Nas aplicações que exploram o espaço bidimensional (2D) as técnicas de HCI já estão consolidadas e incluem as comumente encontradas nas interfaces do tipo WIMP - *Windows, Icons, Menus and Pointing Device*. Através desse tipo de interação os usuários normalmente podem gerenciar os sistemas operacionais, armazenar e gerenciar textos, apresentações, *bitmaps*, áudios, vídeos, entre outros. A interação via dispositivo de apontamento, ou simplesmente *mouse*, é bastante explorada na maioria das interfaces projetadas para *desktops* 2D.

De acordo com Trevisan, um dos aspectos centrais na HCI consiste em como combinar objetos reais e virtuais dentro de um ambiente real, de tal forma que os usuários possam realizar as suas tarefas e interagirem simultaneamente com os objetos reais e virtuais [Trevisan et al., 2002]. Utilizando os métodos já consolidados de projetos de interfaces em ambientes 2D, o usuário será levado a interagir com dois ambientes, um real e outro virtual, o que poderá acarretar numa quebra do seu fluxo natural de interação, quando solicitado ora a interagir com o ambiente real ora com o mundo virtual. Como conseqüência destas dificuldades, os métodos tradicionais não são válidos para projetar e avaliar interfaces em ambientes tridimensionais (3D).

Nas aplicações de Realidade Virtual (VR) técnicas especiais de interação que lidam com o espaço 3D são demandadas. A interação pode ocorrer tanto no sentido usuário-aplicação, quanto no sentido aplicação-usuário. Como exemplos desta última, cenários virtuais utilizados em VR podem dar retorno ao usuário através de dispositivos chamados "hápticos". Neste caso, o usuário sente a reação do mundo virtual e pode reagir de forma diferente dependendo dos estímulos providos pela VR.

Ainda considerando os ambientes de VR, deve ser avaliado se a interação será não-imersiva, ou imersiva. Para os ambientes imersivos as técnicas de interação introduzem novos paradigmas para o usuário, estimulando outros sentidos que normalmente não poderiam ser explorados em ambientes não-imersivos, como por exemplo, a visão estereoscópica [Bowman et al., 2004].

Nos ambientes de Realidade Aumentada (AR) a mudança ainda é mais drástica, principalmente do ponto de vista do usuário, onde os computadores tradicionais com os seus dispositivos e interfaces já bem definidas praticamente não são mais utilizados. Com a introdução de alguns dispositivos especializados, tais como *trackers* e luvas de dados, resolvem-se problemas específicos para algumas aplicações, mas estes dispositivos não podem ser reutilizados ou adaptados em outras áreas de aplicação [Broll et al., 2005].

O *design* de técnicas de interação visa três objetivos principais: desempenho, usabilidade e utilidade. Desempenho diz respeito à quão bem as atividades estão sendo realizadas pelo usuário e pelo sistema, em cooperação, além de eficiência, precisão e produtividade. Usabilidade trata da facilidade em informar o sistema sobre as intenções do usuário, bem como a facilidade de uso e de aprendizado, além do conforto do usuário. Utilidade mostra que a interação ajuda o usuário a atingir os seus objetivos, podendo focalizar na tarefa.

A próxima seção descreve as principais técnicas de

interação para interfaces 3D, apresentadas utilizando a classificação definida por Bowman (2004) para técnicas comumente adotadas em ambientes de VR, além de uma breve discussão sobre a adaptação destas técnicas para ambientes de AR.

3.2. Métodos e Técnicas de Interação

Interação 3D é crucial em sistemas de VR, assim como de AR. Nesta seção são apresentadas algumas técnicas de interação 3D que foram, originalmente, desenvolvidas para ambientes virtuais. É importante ressaltar que a maioria destas técnicas pode ser adequadamente utilizada em ambientes de AR e Virtualidade Aumentada, com poucas ou sem adaptações.

As técnicas de interação para VR foram classificadas em quatro categorias principais, de acordo com a tarefa realizada pelo usuário: técnicas para seleção e manipulação, para controle do sistema, para navegação e de entrada simbólica. Esta classificação foi definida em Bowman (2004), e tem sido amplamente aplicada pela comunidade de interface homem-máquina.

3.2.1. Manipulação 3D

A efetividade das técnicas de manipulação 3D depende das tarefas de manipulação para as quais elas são aplicadas. A mesma técnica pode ser intuitiva e fácil de usar em algumas condições, porém inadequada em outras. É necessário que a interação seja realista, o que significa que o usuário, por exemplo, agarra e move o objeto virtual como manipularia esses objetos no mundo real.

A tarefa de manipulação trata de seleção, ou seja, especificar um ou mais objetos a partir de um conjunto de objetos, para algum propósito. Além disto, o usuário especifica propriedades de um objeto, como posição, orientação, escala, forma, textura e cor. Os objetivos da seleção são consultar um objeto, navegar até este objeto, tornar o objeto ativo e definir a manipulação a ser realizada. Por outro lado, os objetivos da

manipulação são posicionar objetos (*design* de objetos, agrupamento de objetos, *layout* do ambiente virtual), navegar pelo ambiente e realizar uma determinada ação.

Muitas das técnicas de manipulação podem ser usadas em combinação com técnicas de interação para outras tarefas, não diretamente relacionadas com manipulação. Algo a observar é que técnicas de manipulação preservam a forma dos objetos.

As técnicas de interação para manipulação 3D são Apontamento, Manipulação Direta, Mundo em Miniatura, Agregação e Integração e Manipulação 3D para Desktop. Cada uma destas categorias possui várias técnicas associadas. As características destas técnicas são detalhadamente descritas em Bastos (2006).

A Figura 3.1 apresenta a ferramenta DEMEditor, uma aplicação que permite a visualização, manipulação e edição de superfícies 2D/3D construídas a partir de dados de sensoriamento remoto, coletados por radar [Teichrieb e Kelner, 2004]. Esta ferramenta utiliza a técnica Widgets 3D (descrita em [Bastos et al., 2006]) como forma de interação com um dos objetos que compõe o cenário, um holofote, mostrado no canto inferior direito da figura. Quando o holofote é selecionado, o *widget* se torna visível, podendo ser transladado, e sua direção ou área de iluminação pode ser modificada. A Figura 3.1a ilustra o holofote apagado; a caixa branca ao seu redor da Figura 3.1b, c e d significa que o usuário selecionou o holofote e o acendeu. Os retângulos (Figura 3.1b), círculos (Figura 3.1c) e eixos (Figura 3.1d) vermelhos indicam que o holofote foi selecionado no modo de translação, rotação e escala, respectivamente, de forma que o mesmo pode ser transladado para qualquer lugar no ambiente virtual, ou rotacionado em alguma direção ou escalado.

3.2.1.1. Considerações sobre Implementação

Uma questão bastante importante quando da implementação de

técnicas de manipulação 3D é como indicar um evento de seleção ou manipulação. A dificuldade aumenta quando o ambiente possui interseções entre objetos. Por isso, é importante que o sistema forneça algum tipo de retorno para o usuário, que pode ser gráfico, auditivo ou tátil, de forma que o mesmo saiba quando um objeto está selecionado e pronto para ser manipulado.
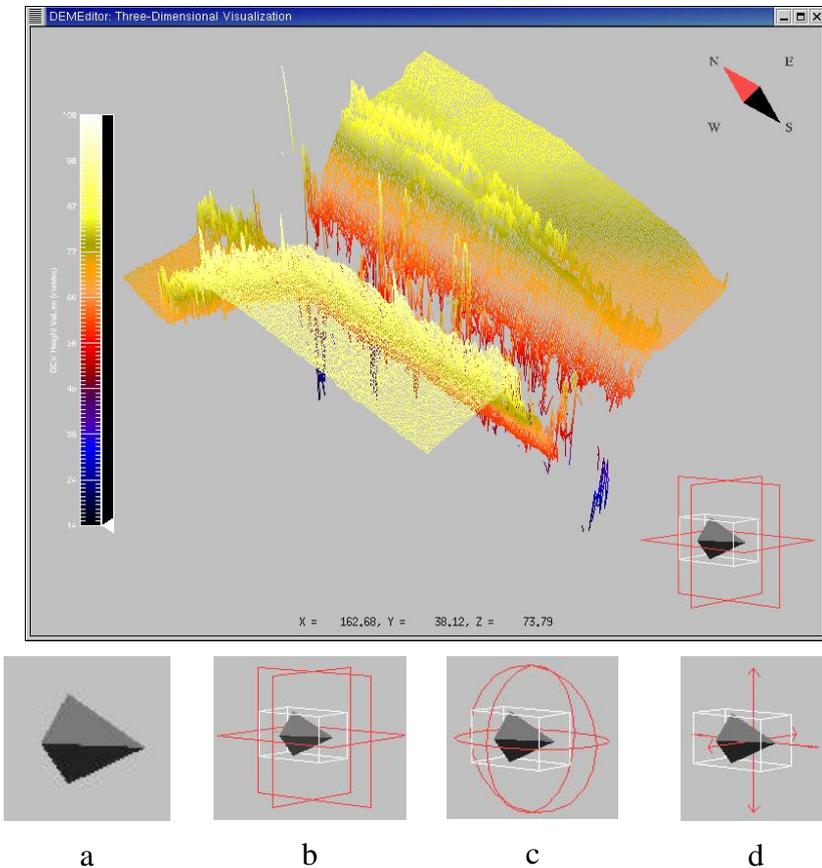
Figura 3.1. DemEditor: técnica Widgets 3D, para manipulação 3D de objetos em aplicações *desktop*. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006)

Exemplos clássicos de retorno visual são o avatar da mão virtual e o avatar caneta, que representam a mão do usuário manipulando objetos na cena, ou então uma caneta que pode ser usada para apontar para o objeto em questão. Uma abordagem bastante utilizada na implementação de técnicas para manipulação é a que especifica que o sistema possui uma lista dos objetos selecionáveis, o que impede a seleção de objetos que não podem ser manipulados pelo usuário.

A integração da técnica de seleção com a de manipulação também é uma questão de implementação importante. Para iniciar a manipulação, a seleção deve ser desabilitada, e o estado da seleção deve ser informado durante a manipulação. Um aspecto da especificação da interface é definir claramente o que acontece no sistema após a manipulação.

3.2.2. Controle do Sistema

Técnicas de interação para controle do sistema servem basicamente para modificar o estado do sistema ou o modo de interação utilizado pelo mesmo. Normalmente, estas ações são realizadas através de comandos disponíveis na interface. Comandos de controle do sistema muitas vezes são integrados com outras tarefas de interação, quando modificam o estado do sistema, ou com todas as outras atividades de interação disponíveis no sistema, quando o usuário os utiliza para controlar o modo de interação a ser utilizado. Um exemplo clássico são comandos acessíveis via menus, como salvar um arquivo, entre outros.

As técnicas de interação para controle do sistema são Menus Gráficos, Comandos de Voz, Comandos de Gestos e Ferramentas. Estas categorias possuem várias técnicas associadas e uma descrição mais detalhada das mesmas encontra-se em Bastos (2006).

A Figura 3.2 ilustra a utilização de menus 2D adaptados para um cenário 3D. Neste caso, o usuário pode utilizar o menu

como está habituado a fazê-lo em aplicações 2D, para diversas tarefas de manipulação de arquivos em um cenário 3D. A figura apresenta um dos menus da aplicação mivaDesk [Teixeira et al., 2007].

Figura 3.2. mivaDesk: técnica Menus 2D Adaptados, para controle de sistemas através de menus gráficos. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006)

3.2.2.1. Considerações sobre Implementação

A implementação das técnicas de controle do sistema deve facilitar o foco na tarefa a ser realizada pelo usuário, evitando que sua atenção seja desviada do objetivo inicial. Além disto, é importante usar uma referência espacial consistente, posicionando corretamente os comandos no ambiente virtual.

Este tipo de técnica de interação é fortemente embasado em entrada multimodal, permitindo o uso de voz, gestos, além dos

menus gráficos posicionados na cena. A utilização destas modalidades variadas de entrada de dados deve ser cuidadosamente avaliada, de acordo com o domínio da aplicação e o perfil do usuário.

3.2.3. Navegação

É sabido que movimentação física tem uma influência positiva nos níveis de presença relatados por usuários quando da interação em ambientes virtuais. É neste contexto que são importantes as técnicas de interação para navegação. O usuário, ao navegar pelo ambiente virtual, pode realizar ações como viajar (explorar) pela cena ou procurar um caminho específico.

Ao viajar (*travel*), o usuário se movimenta entre dois lugares, pela definição da posição (e orientação) do seu ponto de vista. Através desta ação o usuário pode realizar tarefas de exploração do ambiente virtual, de busca por algum local específico do ambiente, e de manobra. No caso da busca, a posição do alvo pode ou não ser conhecida; quando a posição não é conhecida, a busca é chamada ingênua (*naive*), e quando é conhecida e o usuário quer encontrar novamente um local específico diz-se que é uma busca privilegiada (*primed*). Esta tarefa de navegação é amplamente realizada nas aplicações.

A ação conhecida como *wayfinding*, que permite procurar um caminho específico, visa o uso e a aquisição por parte do usuário de conhecimento espacial para definir um caminho no ambiente. O auxílio de pistas e dicas disponíveis no ambiente virtual facilita a realização desta tarefa. Usuários possuem habilidades de orientação diferentes, o que deve ser levado em consideração quando da utilização de técnicas para navegação.

As técnicas de interação para navegação são Locomoção Física, Direcionamento, Planejamento de Rotas, Baseadas em Alvo, Manipulação Manual, Travel-by-Scaling, Orientação do Viewpoint, Especificação da Velocidade e Controles Integrados da

Câmera para Ambientes Desktop 3D. Em Bastos (2006) encontra-se uma descrição detalhada de cada uma destas categorias com suas técnicas associadas, bem como suas características.

A Figura 3.3 ilustra o uso da técnica Zoomback para navegação por um cenário de planejamento de poços de petróleo, gerado pela ferramenta Vis-Petro [Barros et al., 2006]. Esta navegação é realizada escolhendo o ponto de vista desejado no botão Sel da barra de ferramentas lateral, que levará o usuário diretamente ao local definido por aquele ponto de vista. Esta técnica permite explorar o cenário sob diversos ângulos de visualização; na figura o usuário tem uma visão em perspectiva do cenário completo.
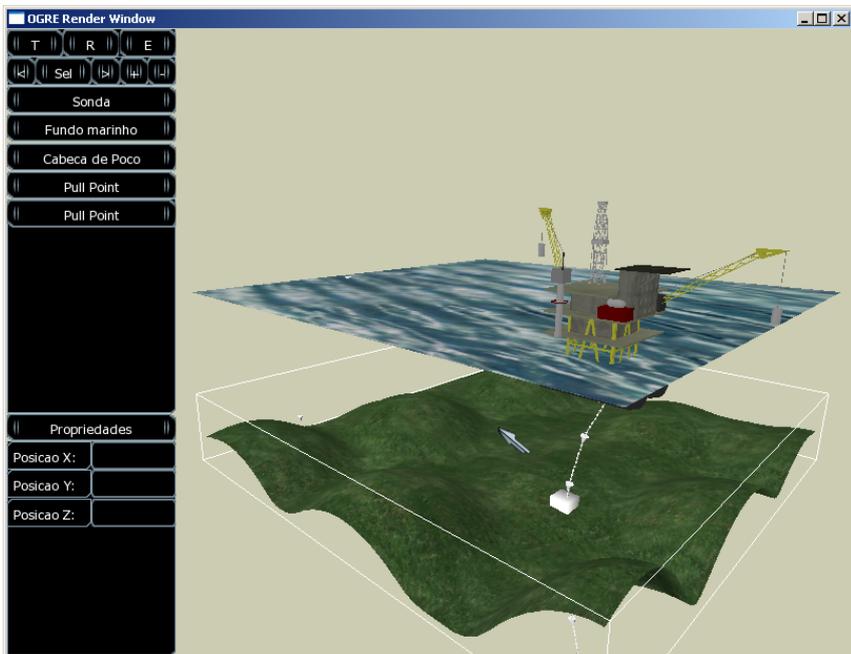
Figura 3.3. Vis-Petro: técnica Zoomback, para navegação baseada em alvo. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006).

3.2.3.1. Considerações sobre Implementação

Existe uma quantidade bastante variada de técnicas de interação para navegação, disponíveis na literatura. Por isso, a simplicidade do uso da técnica deve ser levada em consideração; por exemplo, técnicas baseadas no alvo são simples para se movimentar até um objeto específico, enquanto técnicas de direcionamento são mais apropriadas quando se deseja realizar uma busca.

3.2.4. Entrada Simbólica

Existem diversas situações onde entrada simbólica em aplicações seria útil, tais como deixar uma carta, uma anotação precisa para o projetista em um passeio arquitetônico, entrar nomes de arquivos para operações de abrir/salvar, adicionar nomes a objetos virtuais, especificar propriedades numéricas (por exemplo, largura e posição) e especificar parâmetros em uma visualização científica. Estas aplicações podem ser generalizadas em cenários que envolvem mundos virtuais imersivos ou mundos aumentados. Nestes mundos uma técnica comum de entrada de texto/número é demandada (entrada via teclado, por exemplo), e uma entrada de voz somente não seria suficiente para o usuário interagir com o mundo.

Técnicas de entrada simbólica para interfaces 3D são diferentes de técnicas tradicionais (por exemplo, teclado) pelas diferenças inerentes entre interfaces 3D (não *desktop*) e 2D. Plataformas móveis usam entrada baseada em caneta, na qual o usuário escreve caracteres, símbolos ou outros gestos com uma caneta no dispositivo. Algumas dessas técnicas já foram usadas em interfaces 3D, utilizando este tipo de dispositivo para interação.

As técnicas de interação para entrada simbólica são Baseadas em Teclado, Baseadas em Caneta, Baseadas em Gestos e Baseadas na Fala. Estas categorias e suas características são detalhadamente descritas em Bastos (2006).

A Figura 3.4 ilustra um teclado virtual que aplica a técnica de interação Teclado Soft para entrada de texto na aplicação mivaDesk [Teixeira et al., 2007]. O usuário interage com o mesmo simulando o toque nas teclas de um teclado convencional colocando o cursor controlado por um *tracker* sobre a tecla virtual desejada e fazendo uma postura de seleção com uma luva.

Figura 3.4. mivaDesk: técnica Teclado Soft, para entrada simbólica baseada em teclado. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006).

3.2.5. Técnicas de Interação Específicas para Realidade Aumentada

Técnicas de interação específicas para interfaces de AR foram concebidas, a fim de aproveitar características inerentes deste tipo de interface, tais como a possibilidade de interação do usuário tanto com objetos virtuais quanto reais durante a utilização da aplicação, e a mobilidade.

Atualmente, ainda não existe um consenso, na literatura, sobre como estas técnicas de interação devam ser adequadamente classificadas. No entanto, alguns autores publicaram propostas de classificações, tais como Trevisan (2002), Bowman (2004) e Broll (2005).

Nesta seção serão ilustradas algumas técnicas de interação amplamente empregadas em aplicações de AR, e será traçado um paralelo com suas classificações correspondentes em VR.

3.2.5.1. Interfaces Tangíveis

Da mesma forma que as técnicas de interação para manipulação utilizadas em VR permitem ao usuário selecionar e manipular objetos virtuais, em AR tem-se as interfaces tangíveis. Esta técnica permite a manipulação de objetos virtuais através da manipulação de objetos reais.

Em interfaces tangíveis os usuários manipulam objetos físicos, ferramentas, superfícies ou espaços para interagir com as aplicações. A forma como os usuários manipulam os objetos reais é natural e intuitiva. Em sistemas de AR, os objetos físicos são mapeados usando-se uma função um-para-um com operações sobre objetos virtuais.

Para ilustrar, pode-se assumir como exemplo uma aplicação que disponibiliza um serviço de leilão virtual no qual o usuário interage utilizando um cubo tangível [Teichrieb et al., 2007]. Em leilões reais o arrematador provavelmente desejará ver o lote ou alguns de seus itens antes de finalizar o processo de compra. Ele se sentirá confortável se for possível tocar os objetos com as mãos e observar todos os seus detalhes.

Dessa forma, o serviço de leilão virtual foi desenvolvido permitindo ao usuário interagir com os itens do lote a ser leiloado, visualizando os mesmos em 360°, além de ser possível escalar os objetos para uma visualização mais detalhada, se for o caso.

Rotacionando um cubo de marcadores em sentido horário, o objeto é escalado até o valor mais alto definido na aplicação, e para reduzir seu tamanho o usuário gira em sentido anti-horário. A Figura 3.5 ilustra o usuário manipulando um item do lote, utilizando para isso um cubo tangível cujas faces são marcadores fiduciais.
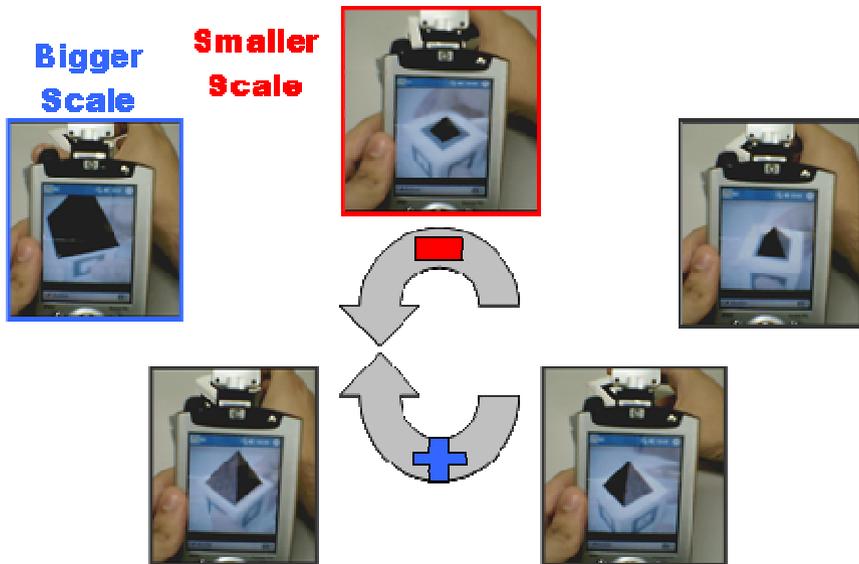
Figura 3.5. Interface tangível para manipulação de objetos. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2007)

3.2.5.2. Interfaces Baseadas em Gestos

Este tipo de interface usa como entrada gesticulações espontâneas do usuário, mímicas e gestos simbólicos. Um exemplo que ilustra esta técnica, quando baseada em posturas (gestos estáticos), é uma aplicação onde o usuário veste um *wearable computer* e uma luva para interagir com a aplicação.

A interação ocorre realizando diferentes posturas pré-

definidas para desempenhar tarefas, tais como abrir e fechar a mão (Figura 3.6). Gestos são identificados pelo uso de uma luva, que detecta diferentes níveis de pressão para todos os dedos, e um *tracker*. A combinação da ativação de sensores de um ou mais dedos se configura num gesto, controlando algum aspecto da aplicação.

Figura 3.6. Interface baseada em gestos para controle do sistema. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006).

3.2.5.3. Walking

A técnica *Walking* simplesmente prevê o andar físico através do mundo 3D, sendo bastante natural para o usuário. Ela é classificada como uma técnica de interação para navegação em ambientes de VR, mas mostra-se muito adequada para aplicações móveis de AR.

Um aspecto importante a considerar é o fornecimento de avisos que auxiliem no equilíbrio do usuário durante a caminhada, de forma a promover entendimento espacial. O caminhar no mundo real não é sempre prático ou factível, uma vez que é limitado por obstáculos espaciais e tecnológicos. Além disto, a margem de movimentação do usuário é diretamente dependente da tecnologia de *tracking* utilizada.

Um exemplo que ilustra o uso desta técnica de interação é uma aplicação de AR que permite ao usuário visitar a galeria de arte de um museu virtual. Para isto, o usuário veste um *wearable computer* (com um *Head Mounted Display* translúcido para

visualização e uma luva com *tracker* para interação) e caminha por um espaço físico [Teixeira et al., 2007]. Uma vez tendo marcadores fiduciais no seu campo de visão, a aplicação reconhece os mesmos e exibe as obras de arte virtuais na cena. A Figura 3.7 ilustra o usuário andando fisicamente pelo ambiente.
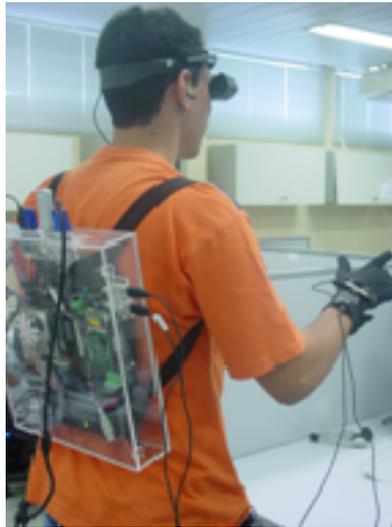
Figura 3.7. Técnica *Walking* para navegação. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006).

3.2.5.4. Reconhecimento de Gestos Pen-Stroke

Esta técnica reconhece o movimento de uma caneta, por exemplo, sobre a tela *touch-screen* de um *handheld* ou *smartphone* (Figura 3.8). Com o uso crescente de dispositivos móveis em aplicações de AR, esta técnica, classificada como técnica para entrada simbólica em ambientes virtuais, tem sido amplamente utilizada em AR.

Figura 3.8. Técnica Reconhecimento de Gestos Pen-Stroke para entrada simbólica. (Imagem cortesia © Grupo de Pesquisa em Realidade Virtual e Multimídia da Universidade Federal de Pernambuco, 2006).

3.3. Considerações Finais

Este capítulo apresentou alguns conceitos relativos a técnicas de interação para ambientes 3D, descrevendo questões relacionadas ao *design* e implementação de técnicas utilizadas em VR e AR. O intuito do que foi exposto é contribuir para uma difusão do conhecimento em áreas onde a literatura ainda é escassa, e ilustrar através de alguns cenários que é plenamente possível adaptar as técnicas já utilizadas em VR para ambientes aumentados.

3.4 Referências Bibliográficas

Schneiderman, B. e Plaisant, C., Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishers, 2004.

Bowman, D. A., Kruijff, E., LaViola Jr, J. J. e Poupyrev, I., 3D User Interfaces: Theory and Practice, Addison-Wesley, 2004.

Trevisan, D. G., Vanderdonckt, J. e Macq, B. (2002) "Analyzing Interaction in Augmented Reality Systems", ACM Multimedia International Workshop on Immersive Telepresence, p. 56-59.

Broll, W., Lindt, I., Ohlenburg, J., Herbst, I., Wittkämper, M. e Novotny, T. (2005) "An Infrastructure for Realizing Custom-Tailored Augmented Reality User Interfaces", IEEE Transactions on Visualization and Computer Graphics, v. 11, n. 6, p. 722-733.

Bastos, N. C., Teichrieb, V. e Kelner, J., Interação com Realidade Virtual e Aumentada, SBC, 2006.

Teichrieb, V. e Kelner, J. (2004) "DEMEditor: a Virtual Reality System to Enhance the Precision of Digital Elevation Models", American Society for Photogrammetry & Remote Sensing, Maryland, American Society for Photogrammetry & Remote Sensing, p. 228-236.

Teixeira, J., Silva, D., Moura, G., Costa, L., Teichrieb, V. e Kelner, J. (2007) "miva: Constructing a Wearable Platform Prototype", Symposium on Virtual and Augmented Reality.

Barros, P., Pessoa, D., Leite, P., Farias, R., Teichrieb, V. e Kelner, J. (2006) "Three-Dimensional Oil Well Planning in Ultra-Deep Water", Symposium on Virtual Reality, Porto Alegre, Sociedade Brasileira de Computação, p. 285-296.

Teichrieb, V., Gomes Neto, S., Farias, T., Teixeira, J., Lima, J., Almeida, G. e Kelner, J. (2007) "Augmented Ambient: an Interactive Mobility Scenario", HCI International.

SEÇÃO B

TECNOLOGIA E PROJETO

Capítulo 4

Ambientes de Hardware e Software para Aplicações de Realidade Virtual e Aumentada

Bruno Barberi Gnecco[1], Marcelo de Paiva Guimarães[1,2] e Rodrigo Damazio[2]

[1] Faculdade de Campo Limpo Paulista
Caixa Postal – Campo Limpo Paulista – SP – Brasil

brunobg@gmail.com

[2] Centro Universitário Adventista de São Paulo
Est. de Itapecerica, 5859 CEP 05858-001 - São Paulo-SP

marcelodepaiva@gmail.com, rodrigo@damazio.org

Abstract

This chapter's objective is to present the most common hardware and software environments that can be used for Virtual and Augmented Reality applications. We begin by presenting and commenting on the use of several hardware systems for processing, such as: personal computers, commodity computer clusters and supercomputers. Then we proceed to present input/output devices, such as head mounted devices, graphical boards, cameras, mice, trackers and gloves, which allow the user to interact with the environment.

Last but not least, we comment on the kinds of software available for Virtual and Augmented Reality: modeling tools, graphical engines, specific frameworks for Virtual Reality and for Augmented Reality and hardware administration systems.

Resumo

Este capítulo tem como objetivo apresentar os principais ambientes de hardware e software que podem ser utilizados para Aplicações de Realidade Virtual e Aumentada. Inicialmente, apresenta-se e discute-se a utilização dos diversos sistemas hardware para processamento, tais como: computadores pessoais, aglomerados de computadores comuns e os supercomputadores. Em seguida, apresenta-se os dispositivos de entrada/saída, como capacetes de visualização, placas gráficas, câmeras, mouses, rastreadores e luvas, que provêm a parte interativa do ambiente para o usuário. Finalmente, comenta-se sobre os gêneros de software disponíveis atualmente para a Realidade Virtual e a Realidade Aumentada, como: ferramentas de modelagem, engines gráficos, frameworks específicos para Realidade Virtual e Realidade Aumentada e sistemas de administração do hardware.

4.1. Introdução

Os cinco sentidos (visão, audição, tato, olfato e paladar) fundamentais do corpo humano formam um conjunto de funções que propicia o relacionamento dos usuários com o ambiente [Aparelhos Sensoriais, 2007]. Por meio deles, o usuário percebe tudo o que o rodeia; e, de acordo com as sensações, interage com o ambiente.

A Realidade Virtual (RV) e a Realidade Aumentada (RA) utilizam os dispositivos de saída para enviar estímulos aos sentidos humanos e os dispositivos de entrada para receber os estímulos dos usuários, o que possibilita a integração dos usuários com os mun-

dos virtuais. Os sentidos são mais utilizados na seguinte ordem: visão, audição, tato, olfato e paladar, apesar do tempo de resposta dos mesmos seguir outra ordem (sabe-se, por exemplo, que o tato e a audição oferecem menor latência neural que a visão, sendo mais adequados dessa forma a estímulos que necessitem de resposta rápida). A RV e RA desejam envolver todos os sentidos do seres humanos de forma combinada. Embora diversos avanços já tenham ocorrido, ainda existem muitos a serem feitos, como, por exemplo, o desenvolvimento de sistemas hápticos mais fiéis à realidade.

Os diversos dispositivos de entrada/saída buscam tornar o ambiente virtual o mais real possível, no caso da RV, e mais natural possível, no caso da RA. Além disso, é importante ressaltar que alguns dispositivos oferecem recursos que os seres humanos não possuem naturalmente, como, por exemplo, os dispositivos hápticos exosqueletais, que habilitam os seres humanos a carregarem mais peso do que podem com seu próprio corpo.

Os ambientes de RV e RA podem ser divididos em quatro partes básicas: um Ambiente Físico (iluminação, ar condicionado, eletricidade); um Sistema Computacional de processamento (*hardware*); um Sistema de Visualização (saída: incluindo vídeo, áudio, retorno tátil etc.); e um Sistema de Interação (entrada: *joysticks*, *trackers*, câmeras etc.). Para a integração das três últimas partes, é necessário um sistema de *software* que una os sistemas de forma apropriada.

Existe um falso conceito destes sistemas serem excessivamente complexos e caros, que não é totalmente verdade nos dias de hoje. Atualmente, diversas aplicações de RA e RV podem ser pesquisadas e desenvolvidas em laboratórios com poucos recursos financeiros, desde que o *hardware* seja capaz de suportar as entradas/saídas e processamento desejados, e que o *software* realize a integração requerida.

Cada aplicação de RV e RA exige certos recursos e podem ser compostas por diversos sistemas distintos. É preciso, então, preocupar-se com todos os seus aspectos, incluindo o ambiente em si (como, por exemplo, com a iluminação, espaço para os usuários e isolação sonora), passando pelo sistema de *software* até os equipamentos de geração de imagens e som.

4.2. Hardware

Pode-se dividir os itens de *hardware* de um ambiente de RV e RA em:

- Ambiente Físico: construção civil, área de interação, iluminação, ar-condicionado, espaço dos usuários etc.;

- Dispositivos de Processamento: responsáveis pela computação, geração de imagens, processamento de entradas etc.;

- Dispositivos de Entrada: responsáveis pelo recebimento das interações dos usuários, como, por exemplo, mouses, luvas e câmeras;

- Dispositivos de Saída: eles que apresentam as imagens, sons e respostas táteis aos usuários, como exemplo, dispositivos de *force-feedback*, som 3D e de visualização de imagens.

4.2.1. Ambiente Físico

O ambiente físico onde o sistema de RV ou RA será instalado deve ser preparado corretamente, de forma que nenhuma limitação do ambiente físico atrapalhe a experiência imersiva do sistema ou seu desenvolvimento. Deve-se estar atento ao local de montagem do sistema. A preparação da sala ou salas usadas pode requerer obras, que são sempre custosas e trabalhosas.

A preocupação inicial é em relação ao espaço físico; este deve ser grande o suficiente para que se possa instalar as telas, mesas de trabalho, os computadores, os projetores etc., sem que se limite a movimentação de pessoas e equipamentos. O ambiente de RV e RA será um espaço de trabalho, e portanto, deve ser confortável para desenvolvedores e usuários. Estações de trabalho com mesas e cadeiras apropriadas devem estar disponíveis para o desenvolvimento, porém sem invadir o espaço destinado ao uso do sistema. Os usuários do sistema podem ficar sentados ou em pé, dependendo dos dispositivos de interação utilizados. O ideal é que exista isolamento sonoro na sala, assim o sistema não sofrerá com ruídos externos. Outro item a ser observado é que o sistema elétrico deve ser suficiente para suprir a potência requerida dos equipamentos.

Além disto, deve se preocupar com a iluminação - o ideal é que o sistema seja montado em um ambiente com luz controlada. Assim, em um sistema de RV, a luz não interferirá nas imagens projetadas na tela; e em um sistema de RA, a iluminação poderá ser corretamente calibrada para que os marcadores sejam detectados pela câmera do sistema sem problemas.

Como os computadores e equipamentos são fontes geradoras de calor, a temperatura ambiente deve também ser controlada, requerendo então o uso de ar-condicionado. Se houver uma quantidade grande de máquinas — como um *cluster*, servidores, roteadores — deve-se pensar em isolar as máquinas em uma sala especial. Além de geram uma quantidade apreciável de calor, eles são ruidosos, e em raras ocasiões é necessário acessa-los fisicamente, de forma que o isolamento é também uma medida de segurança e controle.

Todos os cabos devem estar organizados (força, *genlock*, rede, áudio, dispositivos de entrada, vídeo) para que estes possam

ser identificados e/ou trocados de maneira fácil. Recomenda-se agrupar cabos com cintas e presilhas, e identificar ambas as pontas com etiquetas ou identificadores plásticos. Os cabos devem ser preferencialmente passados em guias apropriadas, e testados antes de sua instalação. No caso de ambientes com uma grande quantidade de equipamentos, sugere-se o uso de pisos elevados ou guias presas ao teto, de forma a facilitar o acesso físico aos equipamentos, já que cabos para ambientes de RV e RA costumam ser bastante longos (verifique as especificações do padrão para ver o comprimento máximo de cada tipo de cabo), sendo também importante que sejam blindados e de boa qualidade, para evitar ruídos.

4.2.2. Dispositivos de Processamento

Durante muito tempo usou-se computadores fortemente acoplados para controlar sistemas de RV. Estes computadores tinham foco em aplicações gráficas e eram projetados especificamente para lidar com computação gráfica, com suporte a *hardware* e arquitetura especializada, como era o caso por exemplo dos sistemas da Silicon Graphics, Inc. Na segunda metade da década de 1990, os computadores pessoais passaram a ser suficientemente poderosos para serem usados para aplicações gráficas complexas, especialmente com a disponibilidade de placas de vídeo aceleradoras, com suporte a primitivas 3D.

Mesmo assim, PCs solitários não eram capazes de substituir supercomputadores completamente; mas, se unidos em rede, com *software* especial para a distribuição dos dados e das tarefas — tornando-se os chamados *clusters* (ou aglomerados) de computadores — seu desempenho era equiparável ou mesmo superior a eles. Com isso, houve nos últimos anos um enorme crescimento do uso de *clusters*, principalmente para as aplicações de RV [Guimarães, 2004].

O princípio por trás do uso de *clusters* está na divisão da tarefa total (a síntese de imagens e simulação do mundo virtual) em outras tarefas menores, que são distribuídas entre os nós, e, quando terminado seu processamento, realizam a sincronização das informações obtidas.

Clusters têm algumas vantagens quando comparados com sistemas fortemente acoplados:

- **Menor custo:** por serem compostos de equipamentos amplamente disponíveis no mercado, *clusters* têm baixo custo quando comparados a supercomputadores. Seu custo de manutenção também é menor, pois não depende de suporte especializado nem reposição de peças customizadas;

- **Melhor escalabilidade:** supercomputadores são projetados para escalarem de uma determinada forma, enquanto *clusters* são mais amorfos e sua escalabilidade é mais simples (em geral, apenas a adição de mais nós e possível reconfiguração do *software*).

- **Maior flexibilidade:** como *clusters* podem ser compostos de acordo com as necessidades do sistema, modificados e atualizados com o tempo, permitindo substituição de apenas algumas de suas partes, são mais flexíveis.

Em compensação, *clusters* são mais difíceis de serem programados, já que não têm uma memória compartilhada global e precisam que os processos executem de forma paralela, transmitindo informações pela rede.

O que diferencia um *cluster* de um grupo de computadores ligados em rede é um sistema de gerenciamento e controle, capaz de prover o que é conhecido como *Single System Image* (SSI). Em outras palavras, o usuário deve enxergar o *cluster* não como uma série de computadores, mas como um único sistema.

Um outro sistema atraente nos dias de hoje são as máquinas multi-saídas. Com o crescimento da capacidade de processamento dos computadores, hoje é possível controlar ambientes de multiprojeção com uma única máquina. Boa parte das placas de vídeo atuais têm duas saídas de vídeo (algumas possuem até quatro), e pode-se usar várias placas de vídeo no mesmo PC. Um PC pode facilmente controlar oito ou mesmo dezesseis saídas de vídeo. Há, obviamente, uma queda de desempenho devido ao compartilhamento de CPUs e barramentos, mas para sistemas que não precisam de grande processamento, é uma solução muito atraente.

Cabe ressaltar que os PCs atuais, principalmente quando possuem placas acelerados de vídeo, são capazes de atender os requisitos de processamento de boa parte das aplicações de RV e RA, sendo ainda que as últimas tecnologias lançadas, como CPUs multi-core, têm proporcioado muitos avanços. Entretato, deve-se estar atento à programação, pois esta deve ser feita de tal maneira que realmente usufrua destes recursos, possibilitando o desenvolvimento de pesquisas em laboratórios com poucos recursos disponíveis.

4.2.3. Dispositivos de Entrada

Para permitir a manipulação e/ou visualização dos mundos virtuais, os usuários utilizam dispositivos não convencionais como capacete de visualização, luvas, trajes, mouse, rastreadores, dentre outros.

O *mouse* e o teclado são os dispositivos mais comuns de entrada de dados em computadores. Têm a vantagem de serem bem conhecidos por usuários de computadores e fáceis de encontrar, configurar e programar. Para a maior parte das aplicações de RV e RA, no entanto, são pouco práticos, provendo uma forma não intuitiva de manipulação do mundo e restringindo o usuário a uma mesa fixa e entradas bidimensionais.

O próximo dispositivo mais comum é o *joystick*. Um exemplo de aplicação que geralmente o utilizará são os simladores de vôo, por assemelhar-se ao controle real de uma aeronave. Há uma infinidade de modelos e tipos - *gamepads*, por exemplo, são muito práticos para navegação em ambientes virtuais que sejam relativamente planos.

Um dispositivo que costuma estar associado à RV é o de luvas com sensores, das quais há vários modelos com sensibilidades a movimentos distintos; a maioria não requer calibragem. Luvas têm a desvantagem de não serem muito confortáveis e provocarem fadiga dos braços, já que os movimentos se tornam mais livres quando os usuários mantêm as mãos à frente do corpo. Outra desvantagem está na necessidade de treinamento dos usuários para usá-las: é preciso memorizar posições e movimentos dos dedos e das mãos. Usuários treinados, no entanto, conseguem manipular objetos virtuais facilmente.

Para suprir estas restrições, foram criados os dispositivos hápticos, nos quais os usuários podem tocar e sentir os objetos virtuais, sendo então especialmente úteis para realçar a informação visual, como por exemplo numa tela de códigos em Braille. Estes dispositivos possuem sensores que monitoram as ações dos usuários e atuadores que aplicam vibrações ou pressões em alguma parte do corpo deles. Eles são capazes de atuar tanto como dispositivos de entrada quanto de saída. Quando atuando como dispositivos de entrada, a princípio têm como objetivo detectar a posição e orientação dos usuários. Como exemplo de tais dispositivos, pode-se citar SideWinder Joystick, WingMan Mouse, Phantom e CyberForce. Seu uso mais comum é em simuladores, como, por exemplo, os de cirurgia e de treinamento médico, que reproduzem texturas, vibrações, inclinação e aceleração sofridas no mundo virtual. Eles geram um altíssimo grau de imersão e interação com o ambiente virtual, mas costumam ser caros e específicos para algumas das aplicações.

Os rastreadores são dispositivos que permitem à aplicação rastrear a posição e/ou orientação tridimensional de partes do corpo do usuário. Há vários métodos usados para o rastreamento: campos eletromagnéticos, sonar, conjuntos de câmeras associados a marcadores reflexivos no corpo, etc. Rastreadores costumam ser caros e precisam ser calibrados, o que é uma tarefa tediosa e por vezes bastante trabalhosa, mas permitem uma interação tridimensional completa.

É possível também rastrear movimentos dos usuários sem que estes usem equipamentos em seu corpo (ou com pequenos marcadores para auxiliar na segmentação das imagens), somente com câmeras dispostas no ambiente. Sistema deste gênero têm a vantagem de ser pouco ou não-intrusivos. Enquanto sistemas simples usam uma única câmera, sistemas mais complexos chegam a usar dezenas de câmeras para evitar oclusão (uma pessoa na frente da outra, por exemplo), que devem ser sincronizadas e calibradas, tornando sua instalação e seu *software* mais complexos.

Uma proposta recente de dispositivo de entrada em ambientes virtuais é o uso de PDAs (*Personal Digital Assistant*). PDAs são pequenos e leves, fáceis de manipular e possuem conectividade sem fios. Suas telas permitem manipular facilmente a simulação, e são fáceis de integrar aos aplicativos existentes, sendo que sua desvantagem consiste na perda de imersão que ocorre quando o usuário move seus olhos das telas do ambiente para a tela do PDA [Guimarães et. al., 2004].

Por fim, outra possibilidade de interação é o uso de reconhecimento de voz, onde comandos ditos em voz alta são processados pelo aplicativo. É uma forma natural, e tem a vantagem de liberar as mãos e braços para outras tarefas, mas as limitações atuais de compreensão de linguagem por computadores fazem-na limitada; há também o problema do ruído ambiente (ou o próprio som emitido pelo aplicativo) interferindo com a captura do áudio.

Um dos dispositivos de entrada mais importante para as aplicações de RA são as câmeras, pois as imagens delas possibilitam a localização dos objetos reais e o alinhamento espacial do mundo real com o virtual. O principal fator que influência o uso das câmeras é sua resolução, sendo também importante notar que o sistema de iluminação do ambiente tem alta influência no processamento das imagens.

4.2.4. Dispositivos de Saída

O sentido humano mais estimulado por ambientes de RV e RA é a visão. Nossa percepção tridimensional é gerada a partir de diversas informações recebidas por nossos olhos, combinadas com informações auditivas, que provêem informação espacial mais limitada. Isso é possível porque esses sistemas utilizam o fato de que cada um dos olhos dos usuários serem afastados um do outro (disparidade binocular), de forma que a imagem de cada olho é ligeiramente diferente da do outro. Estas imagens são combinadas no cérebro e compreendidas com características de profundidade, distância, posição e tamanho. As informações mais importantes são a paralaxe estereoscópica, ou seja, ver perspectivas diferentes com cada um de nossos olhos, e a paralaxe de movimento, que consiste em ver imagens diferentes quando movemos nossas cabeças.

A estereoscopia acrescenta a dimensão de profundidade às telas de projeção dos mundos virtuais e, conseqüentemente, torna-os mais próximos e realistas da forma que os usuários os vêem no mundo real no caso de aplicações de RV, permitindo visualizar objetos 3D no mundo real no caso de aplicações de RA. Enquanto a paralaxe de movimento é gerada pelo aplicativo gráfico, a paralaxe estereoscópica (ou simplesmente estereoscopia) requer suporte do sistema projetor para conseguir apresentar uma imagem diferente para cada olho. Na prática, ilude-se o cérebro humano produzindo artificialmente as duas visões, uma para o olho direito e uma para o olho esquerdo.

Existem diversas formas de prover estereoscopia. Na **estereoscopia passiva**, as imagens do par estéreo são reproduzidas superpondo-se, e utiliza-se algum óculos com um filtro passivo (como polarização da luz, ou separação das cores do espectro) para que cada olho veja apenas uma das imagens.

Na **estereoscopia ativa** os usuários utilizam óculos obturadores (*shutter glasses*), controlados em geral por sinal infravermelho, trabalhando em sincronia com os projetores ou monitores. As lentes alternam entre dois estados, transparentes ou opacas. Quando o projetor exibe a imagem esquerda os óculos fecham a passagem de luz para o olho direito, permitindo que a imagem referente ao olho esquerdo seja somente captada por ele, e vice versa.

Auto-estereoscopia significa obter visão estereocópica sem a necessidade de nenhum dispositivo ligado ao corpo do usuário, como óculos ou capacetes. Existem vários sistemas auto-estereoscópicos disponíveis no mercado, entretanto os principais fatores limitantes desta tecnologia são o custo e a quantidade de usuários que podem visualizar a projeção ao mesmo tempo, além do tamanho da tela. Os monitores com auto-estereoscopia baseiam-se em áreas de visualização que o usuário deve permanecer, que fazem com que uma imagem seja visível para o olho direito e outra para o esquerdo.

Existem diversos modelos de monitores de plasma e LCD disponíveis e com boa qualidade, porém o seu tamanho ainda não é suficiente para ambientes imersivos (aplicações de RV), podendo ser compostos em video-walls para grandes ambientes. Sua limitação principal é não suportar estéreo passivo, e raramente terem uma taxa de *refresh* alta o suficiente para estéreo ativo. Por outro lado, são mais simples de se instalar do que projetores e podem ser utilizados para a maioria das aplicações de RA.

Projetores de boa qualidade são hoje fáceis de se encontrar. Os mais comuns usam tecnologia digital (DLP, ou *Digital Light Processing*) compostos de micro-espelhos que modulam a luz. Os mais caros usam tecnologia CRT (*Cathode Ray Tubes*), a mesma de televisores e monitores convencionais, sendo sua vantagem ter uma varredura mais rápida e suportar resoluções maiores, apesar de serem maiores e mais caros que projetores DLP.

A qualidade da projeção também depende das telas: elas devem ser de um material de cor uniforme, de preferência branca ou bastante clara (prateada ou metalizada). As mais simples são as de acionamento manual, que permitem a fixação do estojo metálico na parede ou no teto, e também permitem o ajuste da altura desejada com múltiplos pontos de parada. As telas especiais com superfícies lenticulares proporcionam ganho na luminosidade da imagem, gerando então uma projeção ótima, mesmo mantendo-se a iluminação do ambiente. No caso em que se utiliza projeção com estereoscopia passiva por polarização, ainda, é necessário utilizar telas especiais que não alterem a polarização da luz nelas projetadas.

Enquanto a visão oferece aos usuários uma percepção do espaço visual à sua frente em um ângulo máximo de 180º, a audição permite a eles a obter informações do ambiente em 360º, além de alguma informação de altura relativa. Tanto a visão quanto a audição compartilham o mesmo objetivo numa simulação, que é o de enganar o cérebro, já que em um sistema de som 3D perfeito não é possível diferenciar realidade e simulação. Quando o cérebro recebe o som vindo de qualquer direção, ele processa as características deste som a fim de determinar e/ou localizar a fonte sonora. A localização espacial é que dá ao som o aspecto tridimensional, sendo por isso conhecido como som 3D ou som espacial.

O som pode agregar diversas informações, como, por exemplo, distância do objeto (pelo volume da emissão), localização

do objeto (pela diferença do som recebido pelos dois ouvidos), tipo de objeto (pelo timbre do som) e tamanho do ambiente (pela sua característica de reverberação). Assim, estimular a audição nas simulações de RV pode aumentar a percepção do ambiente virtual, bem como a imersão. A presença de sons nos ambientes virtuais faz com que o realismo do ambientes seja maior, principalmente com som 3D, que conta com características como posicionamento, reflexão/absorção e geradores sonoros móveis (cálculo do efeito Doppler). Os sons podem ser gerados em alto-falantes ou fones de ouvido, fazendo parte ou não de capacetes.

Apesar de poder existir em um ambiente virtual diversas fontes sonoras, os seres humanos concentram-se em apenas uma fonte de cada vez, dentro de um conjunto limitado de estímulos gerados de maior importância, para as quais a atenção do usuário é desviada. Por isso, para criar ambientes virtuais próximos dos reais é necessário o posicionamento das fontes de som, com as devidas características no espaço 3D.

4.3. Software

Existe uma grande gama de *software* disponível para a confecção, processamento e síntese de ambientes virtuais. Esta seção apresenta uma visão geral dos gêneros de *software* existentes, sem detalhar as particulares implementações.

O capítulo “Ferramentas para Desenvolvimento de Aplicações de Realidade Virtual e Aumentada” descreve as ferramentas em mais detalhes.

4.3.1. Ferramentas de Modelagem

A criação de um mundo virtual requer a modelagem da cena: o ambiente virtual, os personagens, os objetos etc. Para fazer essa modelagem, usa-se programas de modelagem, capazes de manipular a geometria, texturas e preparar animações dos objetos.

Em alguns casos, é possível ainda atribuir ações ou compartamentos aos objetos dentro do próprio programa de modelagem.

4.3.2. Engines gráficos

A síntese de imagens de um aplicativo de RV ou RA é coordenada pelo *engine* (motor) gráfico.

Apesar de *engines* gráficos tentarem ser genéricos, o gênero de aplicativo pode influenciar na escolha do *engine*; jogos costumam ter requerimentos de recursos e desempenho bastante diferentes de aplicativos científicos de visualização de dados, por exemplo.

Entre as características que devem ser buscadas em um *engine*, ressalta-se:

- Suporte à leitura de formatos de arquivos necessários ou integração com *softwares* de modelagem;
- Suporte a texturas e animações;
- Suporte a dispositivos de entrada e saída (sendo que muitos bons *engines* optam por não oferecer suporte algum, obrigando o usuário a utilizar outra ferramenta ou biblioteca que seja mais apropriada para seu aplicativo);
- Existência de efeitos gráficos necessários ou desejados para o aplicativo (como animação de partículas ou sombras);
- Possibilidade de ser executado facilmente em toda a gama de hardware que for ser utilizada para o ambiente virtual;
- Fácil utilização.

4.3.3. *Frameworks* para RV e RA

Com a popularização de aplicativos de RV e RA, *frameworks* específicos começaram a ser desenvolvidos. Esses *frameworks* apresentam funcionalidades específicas para cada tipo de aplicativo, como:

- Suporte a dispositivos de entrada não convencionais (rastreadores, luvas etc.) de forma abstrata, permitindo que novos dispositivos sejam usados sem modificação (ou com poucas modificações) do *software*;

- Suporte a dispositivos complexos de saída, por exemplo com múltiplas saídas de vídeo em disposição arbitrária, algo raramente suportado por *engines*.

4.3.4. *Software* de administração de *hardware*

Sistemas de RV e RA podem se tornar consideravelmente complexos, e sua administração deve ser reduzida ao mínimo. No mercardo praticamente não existem *softwares* que suportem uma vasta gama de dispositivos para RV e RA. Geralmente, utiliza-se o próprio *software* do fabricante ou constroí-se o mesmo no próprio laboratório, conforme os recursos humanos e de *hardware* disponíveis. O ideal é automatizar todo o possível:

- **Iluminação**: é fácil fazer o controle de iluminação por computador hoje em dia, controlando luzes com um mínimo de eletrônica, via porta serial ou paralela.

- **Som**: controle de volume de todo o sistema unificado.

- **Projetores, telas, *switches* de vídeo, teclado e mouse:** ligar, desligar, configurar e chavear por *software*.

- **Disparo de processos:** de maneira simples, de preferência com ícones em uma interface gráfica, sem a necessidade de se pre-

ocupar com o fato da execução não se dar em um aglomerado computacional.

- **Administração dos computadores:** instalação e atualização de programas, configuração, disparo de processos remotos, monitoramento do estado do *hardware* e serviços, devendo ser tudo controlado por um sistema único e de fácil utilização.

4.4. Conclusão

Ambientes de RV e RA têm continuamente melhorado em qualidade. O crescimento da capacidade computacional, a melhora dos dispositivos de visualização, como monitores e projetores e o desenvolvimento de novos dispositivos de entrada (*trackers*, *joysticks*, sistemas com câmeras) ocorrem continuamente. Do ponto de vista de *software*, há um contínuo refinamento dos algoritmos usados, com melhora do desempenho, da qualidade de reconhecimento de padrões em RA e da qualidade de imagem final gerada; há também maior suporte a problemas específicos da área, como renderização estereoscópica ou deformada, suporte transparente a dispositivos de entrada e a computação distribuída.

Com a maior disponibilidade, menor custo e maior uso pelos pesquisadores, a montagem de ambientes para RV e RA tem se tornado mais fácil, já que o desenvolvimento de técnicas, produtos e sistemas por outros pesquisadores é reaproveitado.

Equipamentos de RV e RA estão com custo acessível a qualquer entidade com um mínimo de verba disponível, sendo acessível até mesmo a alguns usuários caseiros. As ferramentas hoje permitem que, com um pouco de criatividade e empenho em desenvolvimento, um sistema original e estado da arte seja implementado, para a realização de pesquisas úteis e inéditas.

A evolução dos sistemas na última década é impressionante - enquanto há dez anos havia pouquíssimas soluções para RV e RA, sempre de alto custo, hoje há uma gama de soluções disponíveis sob licenças de custo baixo ou zero. Essas soluções permitem montar sistemas rapidamente, sem a necessidade de especialistas nem de grandes orçamentos.

4.5. Referências Bibliográficas

"Aparelhos sensoriais: OS SENTIDOS (Integumentary System) (2007). Disponível em http://www.corpohumano.hpg.ig.com.br/apr_sensoriais/apr_sensoriais.html. Consultado em 28/02/2007.

Guimarães, M. P.; Gnecco, B. B.; Zuffo, M.(2004). "Graphical Interaction Devices for Distributed Virtual Reality Systems". ACM Press. Virtual-Reality Continuum and its Applications in Industry. Singapura.pp 363-367.

Guimarães, M. P. (2004) "Um ambiente de desenvolvimento de aplicações de Realidade Virtual baseadas em aglomerados gráficos". Tese de doutorado. Escola Politécnica da Universidade de São Paulo. Dezembro.

Capítulo 5

Desenvolvimento de Ambientes Virtuais

Tereza Gonçalves Kirner[1] e Valéria Farinazzo Martins Salvador[2]

[1] Universidade Metodista de Piracicaba – São Paulo – SP - Brasil

tgkirner@unimep.br

[2] Universidade São Camilo - *São Paulo – SP - Brazil*

valeria@scamilo.edu.br

Abstract

This chapter presents a process of development for virtual environments, based on software engineering concepts as well as on specific characteristics of virtual reality applications. The process follows an iterative approach of software development, which includes a series of steps and phases.

Resumo

Este capítulo apresenta um processo de desenvolvimento de ambientes virtuais, proposto com base nos conceitos de engenharia de software e nas peculiaridades das aplicações de realidade virtual. O processo segue uma abordagem iterativa de desenvolvimento de software, composta por diferentes etapas e fases.

5.1. Introdução

Realidade Virtual (RV) compreende tecnologias avançadas de interface, que permitem ao usuário realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais [Kalawsky, 1993; Kirner and Pinho, 1996].

Uma vantagem importante da realidade virtual sobre outras formas de interação humano-computador, é que o ambiente pode ser visualizado a partir de qualquer ponto de vista, à medida que vão sendo feitas alterações em tempo-real. Comportamentos e atributos podem ser dados a objetos pertencentes ao ambiente, o que propicia a simulação de funções e comportamentos do mundo real enfocado. Para suportar esse tipo de interação, o usuário utiliza dispositivos não convencionais, típicos da área de RV, como capacete de visualização e controle, luvas, *spaceball*, *joystick* ótico, etc. Estes dispositivos dão ao usuário a impressão de que a aplicação está funcionando no ambiente tridimensional real, permitindo a exploração do ambiente e a manipulação natural dos objetos - por exemplo, apontar, pegar, arrastar e rotacionar objetos. [Stuart, 1996; Vince, 2004].

Um ambiente envolve um determinado espaço e uma situação delimitada, incluindo todos os componentes neles inseridos, como o conjunto de objetos e de condições passíveis de serem percebidos e com os quais é possível interagir. Um ambiente virtual é um ambiente interativo, gerado por um computador e disponibilizado através de um sistema de realidade virtual [Stuart, 1996].

De acordo com Martins (2000) e Vince (2004), um ambiente virtual típico deve agregar características que o tornem:

- **Sintético.** Significa que o ambiente é gerado em tempo-real por um sistema computacional. Não é, por exemplo, pré-gravado, como acontece com sistemas de multimídia.

- **Tri-dimensional.** Significa que o ambiente que cerca o usuário é representado em três dimensões (3D) e, além disso, que existem recursos que dão a idéia de que o ambiente possui profundidade e que o usuário pode mover-se através dele.

- **Multisensorial.** Significa que mais de uma modalidade sensorial é usada para representar o ambiente, como sentido visual, sonoro, espacial (de profundidade), de reação do usuário com o ambiente, etc.

- **Imersivo.** Entende-se, aqui, mais do que olhar e ouvir um *display* vindo de um monitor; o *display* necessita criar a impressão de que se está dentro do ambiente produzido computacionalmente. Normalmente, um sistema imersivo é obtido com o uso de capacetes de visualização, mas outros sentidos, como o som e controles reativos, são também importantes.

- **Interativo.** Refere-se à capacidade do computador detectar as entradas do usuário e modificar instantaneamente o mundo virtual e as ações realizadas sobre ele.

- **Realístico.** Envolve a precisão com que o ambiente virtual reproduz os objetos reais, as interações com os usuários e o próprio modelo do ambiente.

- **Com presença.** Caracteriza-se como sendo um sentido subjetivo, responsável por dar ao usuário a impressão de que ele está fisicamente dentro do ambiente virtual.

Como para qualquer sistema de software, a construção de ambientes virtuais requer um processo de desenvolvimento

adequado às suas peculiaridades. Este capítulo apresenta um processo de desenvolvimento que atende as características de ambientes virtuais e aplicações de realidade virtual em geral.

O processo proposto segue uma abordagem de desenvolvimento iterativo de sistemas de software, adaptado às especificidades de sistemas de RV. Esta abordagem combina características de desenvolvimento *top-down* e *bottom-up*, buscando a geração de versões (ou protótipos) que vão sendo sucessivamente refinados, até se chegar a uma implementação que atenda aos requisitos inicialmente identificados [Stuart, 1996; Sommerville, 1997]. As etapas e fases que compõem o processo de desenvolvimento compreendem desde a definição dos requisitos até a implementação e avaliação do sistema.

O desenvolvimento de ambientes virtuais requer um esforço conjugado envolvendo temas relacionados com modelagem gráfica 3D e interface humano-computador (IHC), já que sua construção está ligada ao realismo visual e à interação através de outros sentidos humanos distintos [Martins, 2000]. Assim sendo, o processo ora proposto explora, ao máximo, os recursos de RV e IHC, modelagem gráfica e visualização de informação para validar este processo de desenvolvimento.

Os objetivos principais da abordagem são:

- Permitir a participação dos usuários, para que eles possam experimentar o software que está sendo criado e opinar sobre possíveis melhorias.

- Gerar um sistema que possa ser utilizável e de fácil manutenção, em um período de tempo adequado e a custos aceitáveis.

A realização dos objetivos buscados requer, principalmente:

- Que os projetistas saibam quem serão os usuários, identificando-os em termos de atitudes, comportamentos e necessidades.

- Que a equipe de desenvolvimento atue em conjunto com os usuários.

- Que os protótipos sejam criados rapidamente, avaliados pelos usuários e consecutivamente corrigidos, refinados e melhorados.

O objetivo do presente capítulo é, portanto, apresentar um processo de desenvolvimento de ambientes virtuais, definido e sistematizado através de um conjunto de etapas, fases e atividades. A seção 5.2 descreve o processo de desenvolvimento proposto, a seção 5.3 indica alguns ambientes virtuais desenvolvidos sob a ótica do processo descrito e, por fim, a seção 5.4 apresenta as conclusões do trabalho.

5.2. Processo de Desenvolvimento de Ambientes Virtuais

5.2.1. Exigências

O processo de desenvolvimento de software para aplicações de realidade virtual baseia-se em duas exigências essenciais {Stuart 1996]:

- O ambiente virtual e sua interface devem ser adaptados para a tarefa, isto é, esta exigência está inserida no reconhecimento do fato de que a imersão, no ambiente tridimensional gerado por computador, apresenta oportunidades que não são encontradas na computação gráfica convencional. Para tanto, é necessário que se analisem questões de todo o projeto, projeto da interface e questões importantes de fatores humanos;

- Limites rigorosos de desempenho devem ser cumpridos para que os benefícios de RV sejam alcançados. Isto se refere ao fato de que RV deve ser executada com certa velocidade mínima, a fim de ser utilizável, admitindo-se a taxa adequada de quadros/segundo suficiente para não frustrar o usuário.

Estas duas exigências estão, freqüentemente, em conflito, mas um desenvolvimento bem sucedido de uma aplicação de RV deve atender, simultaneamente a ambos os requisitos. Assim, o desenvolvimento de aplicações de RV deve ter tanto uma abordagem *top-down* (para atender as tarefas totais do projeto), quanto *bottom-up* (para os componentes e desempenho).

5.2.2. Metáforas

A maneira como uma aplicação será implementada dependerá muito de cada aplicação. Entretanto, algumas questões podem ser discutidas levando-se em conta as metáforas em RV que serão adotadas [Martins, 2000].

Metáfora significa a maneira como o usuário se relaciona com o ambiente virtual. A idéia é permitir ao usuário pensar em termos de interação com objetos que estão diretamente relacionados a tarefas disponíveis, em vez de pensar em termos de interação com um computador.

Em RV, o usuário está imerso no ambiente e as metáforas convencionais, tais como interface gráfica do usuário, estão fora do ambiente, não sendo, portanto, apropriadas para tais sistemas.

Construir uma boa metáfora para o ambiente virtual, numa aplicação, é uma tarefa crítica, visto que é necessário saber como esta metáfora determinará a aparência e o comportamento do ambiente e, também, como o usuário interagirá com tal ambiente. Uma boa metáfora permitirá ao usuário interagir eficaz e

confortavelmente com o ambiente virtual para executar as tarefas da aplicação.

É evidente que uma única metáfora não cobrirá todas as aplicações de RV ou todos os domínios, sendo necessário que estas metáforas sejam construídas com base em cada caso. Uma metáfora, também, não deve derivar do mundo real, mas de uma linguagem convencional do domínio da aplicação para que o ambiente seja construído.

Pode-se definir vários níveis de uma metáfora num ambiente virtual:

- **Metáfora(s) para todo o ambiente.** Determina a aparência total do ambiente, incluindo os tipos de objetos que aparecerão no ambiente e tipos de comportamentos;

- **Metáfora(s) para representação de informação.** Como a informação será apresentada ao usuário;

- **Metáfora(s) de interação.** Como o usuário interagirá com o ambiente e com os objetos no ambiente.

Cada nível de metáfora pode conter várias metáforas. Por exemplo, a metáfora de representação de informação pode incluir textos que aparecem no ambiente (talvez numa janela de informação ou texto 3D) assim como a informação mostrada pela cor de objetos. A metáfora de interação pode incluir a manipulação direta de objetos, tais como seleção e botões do menu através de controle.

Experiências com RV e computação gráfica convencional mostram que pode ser interessante mesclar metáforas das duas áreas, facilitando muito a usabilidade do ambiente virtual.

Uma visão geral do processo proposto é apresentada na Figura 5.1. As etapas que compõem o processo são detalhadas a seguir.

5.2.1. Engenharia de Requisitos

Esta etapa tem como objetivo identificar e descrever as necessidades de informação do ambiente virtual pretendido [Sommerville, 1997; Kirner, 2004]. As principais fases e atividades constantes da etapa são:

♦ **Elicitação dos Requisitos.** Envolve ações que visam capturar e registrar as informações necessárias para o entendimento correto e completo das necessidades e expectativas dos usuários Para isso, é importante considerar três elementos essenciais:

- Potenciais usuários do sistema, identificando-se aspectos como: características específicas do usuários, por exemplo, se já possuem experiência com o uso de ambientes virtuais, se a aplicação será mono ou multi-usuário, se os usuários acessarão o ambiente virtual remotamente, se será um ambiente colaborativo, etc.
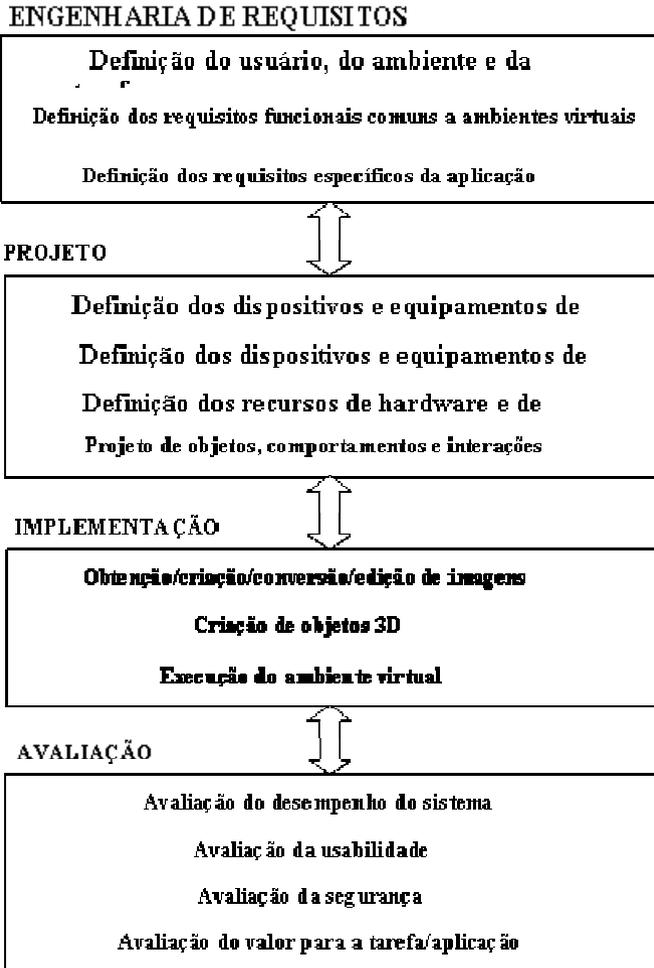
Figura 5.1. Processo de Desenvolvimento de Ambientes Virtuais

- Tarefas a serem executadas no ambiente virtual e como essas tarefas serão realizadas. Por exemplo, os usuários poderão caminhar pelo ambiente virtual, pegar e arrastar objetos, ouvir determinados sons, etc.

- Ambiente da aplicação, que pode ser definido sob diferentes aspectos, como ambiente físico, ambiente de trabalho, ambiente social, etc. Para ambientes virtuais, a definição do ambiente físico em que o sistema funcionará deve merecer atenção especial. Ele pode ser definido com maior ou menor grau de realismo, dependendo da situação, dos usuários e de restrições específicas.

♦ **Especificação dos Requisitos.** Envolve, principalmente, a modelagem e descrição detalhada dos requisitos elicitados [Sommerville, 1997].

- Definição dos requisitos gerais do ambiente virtual, geralmente representados através de modelos conceituais (sob a forma de texto ou por meio de notações gráficas ou formalismos).

- Definição dos requisitos específicos, ligados às características intrínsecas da realidade virtual. Aqui é importante analisar três possíveis alternativas: quando a tarefa a ser executada corresponde a do mundo real; quando a tarefa for executada de maneira diferente do mundo real; e quando a tarefa não corresponde a uma tarefa no mundo real.

- Definição dos requisitos não funcionais, que são relacionados a aspectos de software, de hardware e de fatores externos, que determinam condições ou restrições ao comportamento do ambiente virtual considerado. Deve-se levar em conta tanto os fatores de qualidade exigidos pela aplicação (como usabilidade, portabilidade, confiabilidade, manutenibilidade, etc.) quanto as exigências das aplicações de realidade virtual (como imersão, interação, navegação, sentido de presença, etc.).

Seguindo a abordagem de prototipação, os requisitos levantados e especificados devem ser avaliados e refinados, antes de se partir para a etapa de projeto.

5.2.2. Projeto

Esta etapa tem por objetivo principal, especificar as tecnologias de entrada, de saída e computacionais a serem empregadas no desenvolvimento do ambiente virtual [Martins, 2000; Stuart, 1996].

- **Tecnologias de entrada.** Estas tecnologias, cruciais para sistemas de realidade virtual, podem incluir: capacetes, luvas, câmeras de vídeo, rastreadores de posicionamento, etc.

- **Tecnologias de saída.** Seu objetivo é apresentar informações ao usuário de tal forma que elas possam ser percebidas apropriadamente para a tarefa, dadas as considerações descritas na definição dos requisitos. As tecnologias de saída podem ser: dispositivos visuais, fones de ouvido, dispositivos de tato e força entre outros.

- **Tecnologias de hardware.** Aqui deverão ser definidos quais equipamentos serão empregados, incluindo computadores e dispositivos específicas de realidade virtual.

- **Tecnologias de software.** Um software que atenda aos requisitos de realidade virtual deve ser: interativo, de navegação, de interação, autônomo, que permita o uso de *scripting* e que permita a integração com multimídia.

- **Projeto de objetos, comportamentos e interações.** Aqui deverão ser definidos os objetos, tendo em mente algumas questões como: qual sua geometria, tamanho, escala, cor e textura; os comportamentos que um objeto possa demonstrar; e as interações necessárias para o ambiente virtual, tais como

movimentação, seleção, manipulação, escalonamento, modificação de objetos e de propriedades do mundo, etc.

Em resumo, nesta etapa, são definidas todas as tecnologias a serem empregadas no desenvolvimento do ambiente virtual.

5.2.3. Implementação

Esta etapa tem por objetivo construir o ambiente virtual, compreendendo as atividades descritas a seguir [Martins, 2000; Stuart, 1996]:

- **Obtenção e preparação de imagens.** As imagens necessárias podem ser obtidas através de um *scanner*, câmera de vídeo ou software de desenho. Com a ajuda de um editor apropriado, as imagens devem ser preparadas e convertidas para um formato de arquivo compatível com o software de realidade virtual.

- **Construção dos objetos 3D.** Os objetos geométricos são criados usando-se softwares de modelagem 3D e ferramentas de criação de mundos virtuais.

- **Execução do ambiente virtual.** Consiste em se criar, propriamente, o ambiente virtual. Ou seja, é nesta fase que os objetos 3D são incorporados ao ambiente, assim como os pontos de vista, a luz que iluminará o ambiente, os sensores, a animação, etc.

5.2.4. Avaliação

As principais atividades compreendidas na avaliação são descritas a seguir [Martins, 2000; Stuart; 1996]:

♦ **Avaliação do desempenho do sistema.** Alguns quesitos para se avaliar o desempenho do sistema são:

- Latência total, isto é, o acúmulo de atrasos introduzidos por cada componente do sistema, necessário para gerar uma resposta do sistema como: taxa de dados e de atualização dos rastreadores de posicionamento e tempo gasto para executar a simulação;

- Taxa de atualização do display que é a freqüência com que sucessivas imagens visuais aparecem no display (embora esta taxa contribua para a latência total, deve ser analisada individualmente);

- Robustez que é altamente específica da aplicação, em que suas métricas devem ser aplicadas para cada aplicação;

- Tolerância a falhas que não é uma característica específica de sistemas de RV, mas que é necessária para um emprego real de aplicações, tais como telepresença.

♦ **Avaliação da usabilidade.** Como em outras IHCs, avaliar a usabilidade, incluindo taxa de erro, carga de trabalho, realização da tarefa e aprendizado, pode contribuir para melhorar o projeto. Entre as abordagens mais conhecidas, a avaliação empírica é a mais usada para avaliar a usabilidade de ambientes virtuais. Pode-se utilizar técnicas como análise de *videotape,* entrevistas, questionários e monitoramento psicológico. Através destas técnicas, é possível medir as taxas de erros, carga de trabalho, aprendizado, tempo e taxa de realização de tarefas.

♦ **Avaliação da segurança.** A segurança de software é uma atividade que garante a qualidade de software. Estas atividades tentam encontrar e avaliar casualidades que possam agir negativamente sobre o software e provocar uma falha em todo o sistema. Se as casualidades puderem ser identificadas prontamente no processo de engenharia de software, é possível especificar

características de projeto de software para eliminar ou tratar essas casualidades. A modelagem e análise podem ser usadas como parte da segurança de software. Primeiro, estas casualidades são identificadas. Então, técnicas de análise são utilizadas para determinar a gravidade e a probabilidade de ocorrência. Finalmente, a especificação pode conter uma lista de eventos indesejáveis que devem ser gerenciados (dar respostas desejadas do sistema aos eventos).

♦ **Avaliação do valor da tarefa e aplicação.** Para avaliar o valor de um sistema de RV para a aplicação para o qual foi desenvolvido, é necessário que, primeiramente, defina-se um critério em que o valor será considerado. Desde que a análise da tarefa já tenha sido feita durante o processo de especificação de requisitos, uma abordagem é avaliar o sistema através da análise do sucesso para cada sub-tarefa. Uma falha desta abordagem para avaliar o valor do sistema para a aplicação é que se assume, em alguns casos, erroneamente, que os usuários realizarão tarefas da mesma maneira que antes, apenas um pouco melhor devido ao sistema de RV. Esta abordagem erra quando os usuários podem realizar tarefas com este sistema que não poderiam fazer antes ou de maneira indiscutivelmente diferente da anterior.

5.3. Exemplos de Ambientes Virtuais

Uma série de ambientes e aplicações de ambientes virtuais foram desenvolvidos, valendo-se do processo de desenvolvimento proposto nas seções anteriores. Alguns desses ambientes virtuais são destacados a seguir.

- **Museu Virtual.** É um ambiente virtual que representa um Museu Histórico, de visitação pública, representando uma réplica das salas e objetos existentes no museu [Kirner, 1999].

- **Ambiente de Visualização de Informação.** É um ambiente que combina recursos de realidade virtual e visualização de informação, servindo como uma ferramenta de visualização de dados [Kirner, 2000].

- **Ambiente de Comércio Eletrônico.** Trata-se de um ambiente virtual que segue uma metáfora de um shoping center, para vendas através da Internet [Kirner, 2003].

- **Ambiente Colaborativo para a Educação**. É um ambiente multi-usuário, baseado na teoria pedagógica do construcionismo, funcionando na Internet, para suporte a diferentes aplicações educacionais [Kirner, 2001; Kawamoto, 2006].

- **Máquina de Medir por Coordenadas**. O ambiente virtual simula a máquina, sendo de grande utilidade para o ensino da engenharia na área de engenharia [Calonego, 2004].

- **Ambiente de Simulação de Sistema de Tempo Real.** Trata-se de um ambiente virtual para simulação de uma aplicação na área de manufatura [Kirner, 2005].

- **Visualização de Dados.** É um ambiente virtual, com realidade aumentada, que suporta a visualização de dados em um sistema de pedágio rodoviário [Kirner, 2006].

O uso de um processo sistemático no desenvolvimento dos ambientes e aplicações de realidade virtual teve impactos positivos, em termos de produtividade na execução das etapas e fases e em termos de qualidade do produto final obtido.

5.4. Conclusão

Este trabalho apresentou um processo sistemático de desenvolvimento de ambientes virtuais, baseado nos conceitos de engenharia de software e nas características específicas das aplicações de realidade virtual.

O processo proposto segue uma abordagem iterativa, através da qual se vai criando protótipos, que são avaliados e refinados, visando obter o produto final desejado.

Vários ambientes e aplicações de realidade virtual, alguns incluindo realidade aumentada, têm sido criados com o suporte do processo apresentado. Tal abordagem tem as vantagens de possibilitar um desenvolvimento rápido e mais econômico, e de produzir aplicações de boa qualidade e de fácil manutenção.

Avaliações específicas do processo proposto, envolvendo, por exemplo, estudos empíricos, poderiam indicar mais precisamente os pontos positivos da abordagem e apontar aspectos a serem melhorados.

5.5. Referências Bibliográficas

Calonego Jr, N.; Kirner, C.; Kirner, T.G.; Abackerli, A.J. (2004) “Implementation of a virtual environment for interacting with a numeric command machine”, Proceedings of the 3rd IEEE VECIMS International Conference, Boston, MA, p. 125-130.

Kalawsky, Roy S. (1993) The Science of Virtual Reality and Virtual Environments, Addison-Wesley, Reading, MA, 1993.

Kawamoto, A.L., Kirner, T.G. and Kirner, C.(2006) “Experience on the Implementation of a Collaborative Virtual Environment for Educational Applications”. Proceedings of the XXIX IEEE

International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, p.315-321.

Kirner, C., and Pinho, M.(1996) Introdução à Realidade Virtual, Minicurso JAI/SBC, Recife, PE.

Kirner, T.G., Martins, V.F. (1999) "A Model of Software Development Process for Virtual Environments: definition and a case study". Proceedings of the 2nd IEEE ASSET International Symposium, Richardson, TX, pp. 155-161 32.

Kirner, T.G., Martins, V.F. (2000) "Development of an Information Visualization Tool Using Virtual Reality". Proceedings of the 15o ACM Symposium on Applyed Computing, Como, IT, p.604 – 607.

Kirner, T.G., Kirner, C., Kawamoto, A.L. and Wazlawick, R.S. (2001). "Development of a Collaborative Virtual Environment for Educational Applications. Proceedings of the ACM WEB3D International Conference, Paderborn, Germany, pp. 61-68.

Kirner, T.G., Kirner, C., Aquino Jr., P.T. (2003) "Development of a Virtual Reality-Based Interface to Support E-Commerce Applications" Proceedings of the 3rd IFIP Conference on e-Commerce, e-Business, and e-Government, Guarujá, SP, p.640 – 649.

Kirner, T.G., Martins, V.F. (2004) "Contribuição à Engenharia de Requisitos de Ambientes Virtuais". Proceedings do 7º Workshop de Engenharia de Requisitos, Tandil, Argentina, p.263 – 273.

Kirner, T.G. and Kirner, C. (2005) "Simulation of Real-Time Systems: an object-oriented approach supported by a virtual reality-based tool". Proceedings of the 38th Annual Simulation Symposium, San Diego, p.188 – 195.

Kirner, C., Kirner, T.G. (2006) "A Data Visualization Virtual Environment Supported by Augmented Reality", Proceedings of the XXIX IEEE International Conference on Systems, Man and Cybernetics. Taipei, Taiwan, p.97-102.

Martins, V.F. (2000) Processo de Desenvolvimento de Ambientes e Aplicações de Realidade Virtual. Dissertação de Mestrado (Ciência da Computação), Universidade Federal de São Carlos.

Sommerville, I; Sawer, P. (1997) Requirements Engineering: A Good Practice Guide, John Wiley & Sons, England.

Stuart, R. (1996) The Design of Virtual Environments, McGraw-Hill, Fairfield, PA.

Vince, J. (2004) Virtual Reality Systems, Addison-Wesley, Reading, MA.

Capítulo 6

Ferramentas para Desenvolvimento de Aplicações de Realidade Virtual e Aumentada

Marcelo de Paiva Guimarães[1,2], Bruno Barberi Gnecco[1] e Rodrigo Damazio[2]

[1] Faculdade de Campo Limpo Paulista
Caixa Postal – Campo Limpo Paulista – SP – Brasil

marcelodepaiva@gmail.com, brunobg@gmail.com

[2] Centro Universitário Adventista de São Paulo
Est. de Itapecerica, 5859 CEP 05858-001 - São Paulo-SP

rodrigo@damazio.org

Abstract

This chapter presents an overview of how to use existing tools to develop Virtual Reality and Augmented Reality applications. These tools make development of new applications easier, by abstracting software complexity and reducing development time. We describe the most important features to be considered and sought for in those tools, allowing the developer to make the best choice for his particular needs. We also discuss, for each tool, their main advantages and disadvantages, types of applications they are

better suited for, the developer knowledge required for their use and their future perspective.

Resumo

Este capítulo apresenta uma visão geral do uso de ferramentas já existentes para o desenvolvimento de aplicações de Realidade Virtual e Realidade Aumentada. Estas ferramentas facilitam a criação de novas aplicações, abstraindo a complexidade do software e reduzindo o tempo de desenvolvimento. São abordados os principais aspectos a serem considerados e buscados nelas, o que permite ao desenvolvedor realizar a escolha conforme o seu problema. Também são discutidas as suas principais vantagens e desvantagens, tipos de aplicações a que melhor se prestam, o perfil do desenvolvedor para utilizá-las e suas perspectivas futuras.

6.1. Introdução

A necessidade de ferramentas para o desenvolvimento de aplicações de Realidade Virtual (RV) e Realidade Aumentada (RA) é evidente. Estas bibliotecas devem atender os requisitos básicos de *software*, como: usabilidade, suporte a novas tecnologias, desempenho, manutenibilidade, tolerância a falhas, portabilidade e reusabilidade. Além disso, devem fornecer recursos específicos para as áreas de RV e RA, como suporte a dispositivos de entradas não convencionais (câmeras e mouses 3D) e geração de imagens em tempo real.

Ferramentas de alto nível evitam a necessidade de conhecimento profundo de material de base (como computação gráfica, interface com o sistema operacional etc.), permitindo que o desenvolvedor foque apenas no seu aplicativo em si. Além disso, promovem o reuso de estruturas e algoritmos comuns para gerência dos recursos utilizados. Este capítulo contribui, inicialmente, com

uma visão clara dos principais itens a serem analisados nas ferramentas de desenvolvimento de RV e RA, como, por exemplo, questões de desempenho, escalabilidade, facilidade de uso, expansão, documentação e exemplos disponíveis. Em seguida, as ferramentas mais populares na área serão avaliadas e comentadas, incluindo: Blender, Panda3D, Ogre, ARtoolKit e ARtoolKit Plus, entre outras. Cada uma das ferramentas apresentadas será analisada conforme os itens desejáveis em tais ferramentas. Para que o leitor tenha uma noção do que cada uma delas é capaz. Ao terminar a leitura deste capítulo, o leitor estará apto a fazer uma escolha consciente da ferramenta mais apropriada para sua pesquisa.

A Figura 6.1 mostra uma visão geral da hierarquia das principais ferramentas de desenvolvimento. Esta hierarquia é composta de: sistemas operacionais; bibliotecas de comunicação e sincronização (utilizadas somente em aplicações distribuídas); bibliotecas e pacotes gráficos (coleção de classes que oferecem um conjunto de serviços); e pacotes desenvolvimento (conjunto de *softwares* e demais artefatos usados para criar as aplicações). Nota-se que existe uma vasta gama de opções disponíveis para cada um dos graus da hierarquia, e que essas opções podem ser combinadas para criar um aplicativo. Todas as opções podem ser utilizadas tanto para as aplicações de RV quanto de RA. Quanto mais alta na hierarquia, a ferramenta oferece maior abstração dos detalhes de desenvolvimento. A maioria dos projetos atuais tem utilizado os pacotes gráficos, pois requerem menos esforço para o desenvolvimento, em comparação com o uso direto de APIs como OpenGL ou DirectX, além de serem mais flexíveis, possibilitando a customização conforme os requisitos do projeto.

Quanto aos pacotes de desenvolvimento de conteúdo, estes evoluíram muito nos últimos anos. A solução livre mais comumente usada, o Blender, oferece além dos serviços de modelagem recursos de interação via linguagem de programação

Python, além de ferramentas para criação de aplicativos simples sem o uso de programação. Em geral, contudo, é preferível utilizar ferramentas externas para a criação do aplicativo propriamente dito, devido à maior flexibilidade e desempenho. Atualmente, tornou-se comum todas as ferramentas de desenvolvimento fornecerem *wrappers* para linguagens interpretadas como Python e Lua, permitindo que o desenvolvimento utilizando essas linguagens seja feito dentro de um ambiente de alto desempenho e sem as limitações de aplicações de criação de conteúdo.
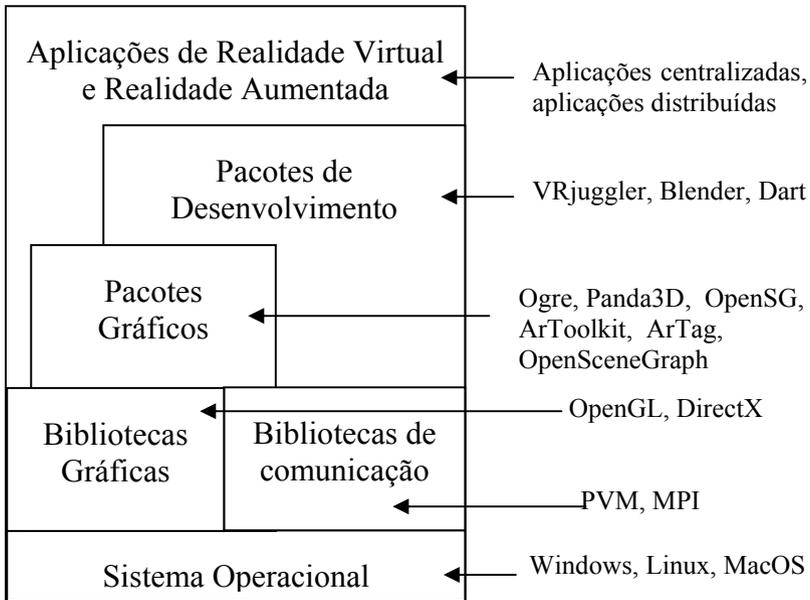
Figura 6.1 Hierarquia das ferramentas de desenvolvimento.

6.2 Aspectos gerais

Existem fatores desejáveis nas ferramentas de desenvolvimento de aplicações de RV e RA, como: facilidade de uso, confiabilidade,

desempenho, legibilidade, robustez, escalabilidade, expansibilidade, documentação, compatibilidade e reusabilidade do código que as utiliza. Esses fatores devem servir como requisitos e preocupações a serem avaliados durante o processo de escolha de qual ferramenta que será utilizada em um determinado projeto.

Para atender a esses requisitos e preocupações, são necessárias ferramentas de desenvolvimento que acompanhem a rápida evolução a que estão sujeitas as áreas envolvidas. Estas ferramentas devem direcionar o desenvolvimento de aplicações de alta flexibilidade e qualidade, a fim de que possam adequar-se com mais facilidade ao dinamismo das tecnologias e dos ambientes onde são utilizadas. É difícil encontrar uma ferramenta que englobe todas as características desejáveis, pois, geralmente, cada uma delas é mais apropriada para um certo tipo de aplicação, levando à necessidade de uma escolha cuidadosa e consciente por parte do desenvolvedor. Uma escolha comum, por exemplo, é a decisão entre uma ferramenta de uso simples com recursos limitados e outra de uso mais complicado, porém que oferece mais recursos. No desenvolvimento de aplicações de RV e RA, essas preocupações também devem estar presentes, e os principais requisitos que um ambiente de desenvolvimento deve fornecer são [Guimarães, 2004; Rocha, 2003]:

• **Desempenho:** a ferramenta deve fornecer recursos que possibilitem alto desempenho das aplicações: caso contrário, pode causar baixa interatividade e falta de imersão. Além disto, o baixo desempenho do sistema pode causar uma baixa taxa de apresentação dos quadros, o que pode gerar problemas conhecidos como *cybersickness* (dores de cabeça, náusea e instabilidade postural) aos usuários. Um desempenho suficiente do subsistema gráfico é fundamental;

• **Extensibilidade:** como novos recursos surgem constantemente, é importante que a ferramenta possibilite a extensão das funcionalidades já disponíveis. Viabilizando que, por exemplo, um novo tipo de rastreador ou *joystick* seja usado nas aplicações ou que novos recursos de uma placa gráfica mais moderna sejam incorporados sem modificação do código do aplicativo;

• **Flexibilidade:** a ferramenta deve permitir a alteração de parâmetros das aplicações em tempo de execução; fácil incorporação ou modificação de funcionalidades. Assim, as últimas tecnologias, como algoritmos de reconhecimentos de imagens específicos ou efeitos gráficos que necessitem de *hardware* de última geração, podem ser incorporados facilmente, e simplesmente desativados quando não for possível utilizá-los;

• **Simplicidade:** embora as aplicações de RV e RA sejam naturalmente complexas, pois envolvem conhecimento de diversas tecnologias, o ambiente de desenvolvimento não precisa sê-lo. O ambiente deve possibilitar a criação de aplicações da forma mais fácil possível, permitindo então o desenvolvimento sem o conhecimento de todas as particularidades do ambiente. Por exemplo, um editor de cenários, sistema de gerenciamento de objetos e o suporte a linguagem *scripts* podem auxiliar muito no processo de desenvolvimento de um jogo, fazendo com que a lógica e conteúdo do jogo sejam isolados de sua infra-estrutura;

• **Robustez:** as aplicações de RV e RA são complexas, o que aumenta a possibilidade de ocorrência de falhas, sejam elas de *hardware* (defeito de uma placa, queda da rede, queda de um nó) ou de *software*, como bugs, *deadlocks*, exceções e falta de recursos. Por isso, é necessário que o ambiente forneça recursos para o tratamento de falhas, permitindo que as aplicações continuem executando mesmo se algumas destas ocorrerem;

• **Portabilidade:** em sistemas que necessitem ser executados em múltiplas plataformas, como é comum no caso de jogos, esta característica pode ser um requisito importante. Ela consiste na possibilidade do ambiente fornecer recursos para que as aplicações de RV e RA possam ser instaladas e facilmente portadas para diferentes arquiteturas de *hardware* (*Personal Computers* — PCs, *Personal Digital Assistant* — PDAs, consoles de jogos) e de *software* — sistemas operacionais (Linux, Irix, Windows) e linguagem (C/C++/C#, Java). Todavia, outras características podem ser consideradas determinantes de portabilidade, como suporte a dispositivos, a habilidade de abrir e converter uma variedade de formatos de entrada e outros. Mesmo nos sistemas que não são planejados para múltiplas plataformas, é recomendável um mínimo de consideração desse fator, caso ele venha a ser um requisito;

● **Processos-leves seguros (*Thread safety*):** suporte de forma simples e automática a processos-leves, que não exige um trabalho ou a utilização de outras soluções em vez de processos-leves. Esse recurso é especialmente importante para que os aplicativos possam tomar vantagem da utilização de CPUs com múltiplos núcleos, que vêm se tornando comuns tanto em PCs quanto em consoles de jogos;

● **Heterogeneidade:** se possível, é desejável que a mesma aplicação possa ser executada em máquinas heterogêneas. Assim, pode-se, por exemplo, ter alguns usuários utilizando a mesma aplicação do Windows, enquanto outros no Linux, e que ambos obtenham a mesma experiência, além de poderem interagir uns com os outros sem diferenças.

Um ambiente de desenvolvimento ideal seria aquele que além de resolver os problemas pertinentes às aplicações de RV e RA atendesse os requisitos básicos de *software*. Visando

principalmente o uso comercial, em que os ambientes geralmente não são de alta qualidade como os dos laboratórios de pesquisas, esses requisitos tornam-se características fundamentais para uma ferramenta de desenvolvimento de tais aplicações.

6.3 Ferramentas de Realidade Virtual

Todas as ferramentas voltadas para RV podem ser utilizadas para RA. Geralmente, as ferramentas de RA oferecem também recursos adicionais de Visão Computacional, para que possam se integrar facilmente com o ambiente.

6.3.1 Panda3D

Panda-3D é um *engine* para jogos desenvolvida pela Disney e pela Universidade Carnegie Mellon. Um *engine* é uma coleção de módulos de simulação que não especifica diretamente o comportamento ou ambiente do jogo. *Engines* incluem recursos como, por exemplo: captura de eventos de entrada; geração de saída gráfica e de áudio; gerencia dinâmica do mundo de jogo; facilidades para criação de lógica da aplicação. Panda-3D permite o desenvolvimento de aplicações 3D e de jogos. Ela foi desenvolvida em C++, mas fornece um *binding* para a linguagem Python, uma linguagem interpretada, interativa, tipada dinamicamente e fortemente, orientada a objetos, e pensada e estruturada para o ensino de programação. Por isso, tem uma compreensão simples da estrutura para o fácil aprendizado (baixa curva de aprendizagem) [Netto et al., 2006].

O processo de desenvolvimento com o Panda3D consiste em escrever programas em Python para controlar esta biblioteca, e a programação se baseia na manipulação de cenários e atores pré-criados. Utilizando Panda 3D é possível navegar em cenas 3D, utilizar saída estereoscópica, utilizar diversos dispositivos de interação, como capacetes de visualização (HMD – *Head- Mounted*

Display), sensores de posição e outros. Cabe ressaltar que esta biblioteca é de código livre http://panda3d.org/. Para a modelagem das peças e do ambiente pode-se utilizar ferramentas de modelagem como o Blender e o 3DStudio Max.

Além dos recursos para manipulação de imagens e ambientes desta biblioteca, como a capacidade de adição de estereoscopia, destaca-se a baixa curva de aprendizagem, o rápido desenvolvimento de ambientes, a fácil instalação, a documentação e os exemplos disponíveis, e o suporte dos sistemas operacionais Windows e Linux.

6.3.2 Ogre

OGRE (*Object-oriented Graphics Rendering Engine*) é um *engine* gráfico que funciona em Windows, Linux e MacOS X, e trabalha tanto com OpenGL como DirectX. Ele provê uma vasta gama de *plugins*, ferramentas e *add-ons*, que favorecem a criação de vários tipos de aplicações gráficas, empregando diversos conceitos inerentes ao desenvolvimento destas. Este *software* é licenciado sob os termos da GNU Lesser Public License (LGPL) e pode ser encontrado em http://www.ogre3d.org/ [Farias et al., 2006; Netto et al., 2006].

Foi desenvolvido em C++ e designado a tornar a implementação de aplicações mais fácil e intuitiva para os desenvolvedores, utilizando gráficos 3D acelerados por *hardware*. Ele possibilita a utilização de diversas configurações de *hardware* 3D (GPUs) disponíveis no mercado, desde as mais obsoletas às mais sofisticadas.

O OGRE fornece uma interface que permite que o desenvolvedor verifique se o *hardware* disponível tem capacidade mínima para executar a aplicação. Isto possibilita o ajuste de uma

técnica alternativa a ser utilizada para que se tenha um melhor proveito dos recursos disponíveis. Este ajuste de técnica também pode ser aplicado aos materiais e efeitos, incluindo versões de *shaders* e texturas multicamadas.

Esta biblioteca possui uma boa documentação e está bastante difundida pela comunidade. A interface nativa é em C++, mas existem *wrappers* para Java, .Net e Python, o que facilita a sua utilização em diversos ambientes.

OGRE não é um *game engine*, mas um *rendering engine* genérico que pode ser incorporado a bibliotecas de tratamento de entradas, de processamento de som e as plataformas que disponibilizem algoritmos de inteligência artificial.

6.3.3 OpenSceneGraph

O OpenSceneGraph (OSG) é uma biblioteca de gerência de cena baseada em grafos em cena, que implementa vários algoritmos de otimização: descarte por campo de visão, plano de oclusão, descarte de pequenos objetos, suporte a níveis de detalhe discretos, ordenação por estado, suporte a diversos tipos de arquivos através de uma interface plugável, *shaders*, *vertex arrays*, *display lists* e *vertex buffers*, sendo os três últimos otimizações específicas do OpenGL [Silva et al, 2003].

A biblioteca foi projetada com uma visão clara de engenharia de *software*, utilizando diversos padrões de projeto para permitir que a biblioteca seja facilmente expansível e que suporte todos os recursos que podem ser necessários a aplicações de RV.

Justamente devido a essa expansibilidade, o OSG oferece uma vasta gama de *plug-ins* que permitem integrá-lo facilmente com outras bibliotecas, tais como OpenDE para simulação física, OpenAL para som e a OSGART (disponível em duas versões, uma

livre e outra comercial), que oferece funções específicas para RA. Além disso, muitas *engines* foram criadas por cima do OpenSceneGraph, como é o caso da Delta3D e Cooki3D.

Apesar da grande expansibilidade, o OpenSceneGraph é preso com a API OpenGL, de forma que não pode ser usada em algumas plataformas (notadamente Xbox 360). Em todas as plataformas onde o OpenGL é suportado, como Windows, Linux, MacOS X, Playstation 3, Irix e outras, o OpenSceneGraph pode ser utilizado diretamente, inclusive com suporte a renderização em processos-leves, de forma a não bloquear o processamento principal do aplicativo devido à comunicação com o *hardware* 3D.

Finalmente, vale notar que a biblioteca acompanha uma grande quantidade de exemplos para demonstrar todas as suas funcionalidades, e um projeto de ampla documentação está em progresso. Disponível em http://www.openscenegraph.org/.

6.3.4 OpenSG

Desenvolvida por diversos grupos de pesquisa, sendo a maioria da Alemanha, a OpenSG é uma biblioteca que utiliza grafo de cena para a geração de imagens em tempo real. O seu principal objetivo é fornecer um ambiente de manipulação de estrutura de dados em múltiplos processos-leves seguros (*multithreaded-safe*), que não comprometa o desempenho. Para isso, o controle dos processos-leves é realizado de maneira explícita, ou seja, os desenvolvedores devem se preocupar com os travamentos (*lock*) e destravamentos (*unlock*) dos processos-leves, o que dificulta o uso desta biblioteca. A OpenSG fornece recursos para que um grafo de cena seja manipulado por processos-leves independentes [Guimarães, 2004].

Esses processos-leves geram a necessidade de replicação de dados, mas o esquema de organização de dados (dados separados

da estrutura da cena) faz com que apenas os dados necessários sejam replicados. Além disso, cada uma delas possui uma lista de atributos alterados. Esta lista é utilizada para que a sincronização seja feita apenas dos dados que foram alterados.

OpenSG possui diversos recursos para a criação de objetos, sendo possível construir um vetor com a indicação de como as estruturas devem ser criadas, como por exemplo, triângulos, linhas, pontos e malhas de triângulos. Além disso, permite a reorganização dos dados por intermédio de índices.

Essa biblioteca fornece a possibilidade de ser estendida. Por exemplo, pode ser utilizada para escrever novos leitores de formatos de imagem. Atualmente, ela disponibiliza leitores de VRML e OBJ.

O OpenSG tem como base um conjunto de bibliotecas básicas, que são compostas por funções de baixo nível e classes bases, por exemplo, vetores e volumes. Ainda, contém uma abstração do sistema operacional e empacotadores (*wrappers*) básicos para os múltiplos processos-leves. Além das bibliotecas básicas, possui as bibliotecas de sistema, que contém os objetos de alto nível, como, os métodos de manipulação de grafos de cena, os leitores de imagens e cenas, e as classes de manipulação de janelas. As aplicações OpenSG podem ser executadas atualmente nos sistemas operacionais Windows, Linux, MacOSX, Solaris e outros. Pode ser encontrada em http://www.opensg.org/.

6.3.5 Crystal Space

Crystal Space é um *engine* para o desenvolvimento de aplicações 3D escrito em C++. Ele é um *software open source* que suporta Microsoft Windows, Linux, Unix e Mac OS X. Sua documentação é boa, com exemplos de suas capacidades básicas, e há um editor

de cenários. Destaca-se nesta ferramenta a modularidade de seus componentes, visto que funcionam de maneira independente uns dos outros. Seu site é http://www.crystalspace3d.org/.

Crystal Space disponibiliza diversas funcionalidades, como: suporte a OpenGL (em todas as plataformas), SDL (em todas as plataformas SDL), X11 (UNIX e Linux) e SVGALib (Linux); mapeamento de texturas com perspectiva de sombreamento; mipmapping; precisão de sub-pixel e sub-texel ; uso de portais para delimitação de áreas permitindo renderização seletiva da geometria do ambiente; iluminação estática com sombreamento pré-calculado; iluminação dinâmica colorida com sombreamento transparente e semitransparente; espelhamento; suporte a árvores BSP; suporte a superfícies curvas; suporte a correção de profundidade e neblina volumétrica colorida; sistema hierárquico de detecção de colisão; teste de visibilidade baseado em *z-buffer*; suporte a sons 3D; movimentação de objetos e *scripts* para controle da movimentação; suporte a LOD.

6.3.6 Irrlicht

Irrlicht é um *engine* 3D escrito em C++ de alto desempenho e licenciado sob a licença zlib. Possui uma boa documentação, com diversos exemplos, e implementa diversas características desejáveis em um *engine* moderno, como sombras dinâmicas, sistemas de partículas, animação de personagens, suporte a cenas internas e externas e detecção de colisão. Suporta tanto Direct3D quando OpenGL, e roda em Windows, Linux e MacOs X.

Outras características importantes desta biblioteca incluem: uma larga e extensível biblioteca de materiais, incluindo *vertex* e *pixel shaders*; *binding* para linguagens .NET, renderização por *software* para plataformas sem aceleração gráfica, API nativa para GUI 2D (botões, listas etc.) e funções de desenho 2D, suporte a

uma larga gama de formatos de modelos e imagens, e outros. A biblioteca possui boa documentação e exemplos. Pode ser baixado de http://irrlicht.sourceforge.net/

6.3.7 VRJuggler

VR Juggler é um *framework* que esconde a complexidade da infra-estrutura e das especificações dos nós (dispositivos, monitores e computadores). As aplicações são independentes dos dispositivos de entrada/saída. Os desenvolvedores podem acessar diretamente as API gráficas (OpenGL ou OpenGL Performer) e utilizar diversos dispositivos como luvas, mouses, monitores e capacetes de visualização. Foi projetado como uma solução padrão para aplicativos de RV [Guimarães 2004]. Seu site é http://www.vrjuggler.org.

6.3.8 Blender

Blender é um *software* (editor gráfico) de modelagem e animação 3D, que está sob a licença GNU-GPL. Ele oferece uma gama enorme de recursos e funcionalidades para criação de objetos 3D, renderização de cenas, animação com um editor de vídeo integrado e pós-produção, tudo em um único ambiente de trabalho. Ele está disponível para várias plataformas, como Microsoft Windows, Mac OS X, GNU/Linux, IRIX, Solaris, FreeBSD, SkyOS e MorphOS. Esta ferramenta high-end fornece funcionalidades similares às encontradas em *softwares* comerciais como XSI, Cinema 4D, 3D Studio Max e Maya. Estas características incluem avançadas ferramentas de simulação tais como dinâmica dos corpos rígidos, dinâmica de fluidos, dinâmica de corpos flexíveis, avançadas ferramentas de modelagem, poderosas ferramentas de animação de personagens, e sistema de materiais baseados em Node e um *wrapper* para a linguagem Python [Brito 2006, Alves 2006].

Apesar do arquivo nativo do Blender ser .BLEND, ele fornece recursos para leitura e escrita der muitas extensões tais como .3DS, .DXF, .WRL (VRML 2.0, VRML 95), .X (Directx), .MD2, entre muitas outras. Os avanços realizados nos últimos anos permitem que o Blender seja comparado com *softwares high-end* comerciais, tais como Maya e 3D Studio Max. A documentação sobre ele é bastante farta. Disponível em http://www.blender.org.

6.3.9 FluxStudio

Flux Studio (conhecido como Vizx3D) é uma ferramenta gráfica (editor) de modelagem e animação voltada para o desenvolvimento de aplicações para Web (3D) e conteúdo de mundos virtuais. Ela suporta apenas Windows. A licença deste *software* pertence à Media Machines. Como é um editor gráfico, é de fácil utilização e é orientado a visualização, sendo capaz de abrir conteúdos X3D, VRML97 e vários outros formatos de arquivos.

Utilizando a sua interface gráfica é possível definir animações, regras de interatividade que podem disparar luzes, sons e animações. A GUI também fornece recursos para depurar as aplicações e *wizards*. Disponível em http://www.mediamachines.com/.

	Tipo	Documentação	Fácil de usar	Recursos	Custos	Linguagem de Programação
Panda3D	API, GUI	++	++	++	+++	C++, Python
Ogre	API	+++	+	+++	+++	Java, .Net, Python
OpenSceneGraph	API	+	++	+++	+++	C++, Lua, Python, Java, .NET
OpenSG	API	+++	+	++	+++	C++, Python
Crystal Space	API, GUI	++	+	+++	+++	C++
Irrlicht	API	++	++	+++	+++	C++, .NET
VRJuggler	GUI, API	+++	+	++	+++	C++
Blender	GUI	+++	+	+++	+++	Python
FluxStudio	GUI	+++	+++	++	-	Não oferece

Tabela 6.1 - Tabela de comparação de ferramentas de RV.

6.4 Ferramentas de Realidade Aumentada

As ferramentas voltadas para RA auxiliam a construção dos objetos virtuais e sua integração ao ambiente real, incluindo alguns comportamentos, como, por exemplo, quando um marcador é detectado um certo objeto é adicionado na cena. A seguir são apresentadas algumas ferramentas voltadas para RA. Além destas, existem outras como o ARStudio, JARToolKit e o DART.

6.4.1 ARToolkit

O ARToolKit é um *software* para a detecção dos marcadores nas imagens e adição dos objetos virtuais nas imagens do mundo real. Este *software* utiliza métodos de visão computacional para fornecer a posição e o rotação 3D de padrões em imagem do ambiente. Estes

padrões são expressos por marcadores 2D e são previamente cadastrados antes da execução da aplicação.

As seguintes etapas são executadas para a detecção dos marcadores: inicialmente, a imagem capturada pela câmera ou por outro processo é transformada em uma imagem binária (com valores em preto e branco). Logo após, o *software* analisa a imagem e encontra os marcadores, e os compara com os previamente cadastrados. Quando encontrado um marcador, um objeto tridimensional virtual é adicionado à imagem real, na posição e orientação do marcador original.

As principais características do ARToolKit são: *tracking* para posicionamento e orientação de uma câmera; marcadores que têm como padrão um quadrado de bordas pretas; possibilidade de uso de qualquer marcador, desde que o mesmo obedeça ao padrão; código simples para calibração da câmera; bom desempenho para aplicações de RA em tempo real; distribuição multi-plataforma (por exemplo, SGI IRIX, Linux, MacOS e Windows) e distribuição do código fonte completo, que pode ser encontrado em http://www.hitl.washington.edu/artoolkit/.

6.4.2 ARToolkit Plus

O ARToolKitPlus baseia-se no ARToolkit. Esta ferramenta fornece algumas otimizações, como, por exemplo, possibilidade de se utilizar computações de ponto fixo ao invés de ponto flutuante, com o objetivo de gerar aplicações eficientes para dispositivos móveis, tais como PDAs e *smartphones*. O sistema de marcadores é semelhante ao do ARToolKit, porém o desenho no interior do marcador é a sua identificação. Esta codificação possibilita que o usuário utilize até 512 marcadores diferentes, facilitando a identificação dos marcadores (diminui a incidência de confusão de identificação).

Destaca-se no ARToolKitPlus a utilização da técnica de limiar adaptativo, que permite o ajuste dos sistema de detecção dos marcadores conforme a luz do ambiente sofre alterações. Disponível em http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php.

6.4.3 ARTag

O ARTag também teve como base o ARTookit. Seu objetivo foi resolver alguns problemas encontrados nesta última ferramenta, principalmente no processo de detecção de marcadores, tais como o falso positivo — quando o sistema acusa a presença de um marcador, mas ele não existe; falso negativo — quando o sistema não acusa a presença de um marcador, mas ele existe; e o de confusão — quando o marcador no ambiente é um e o sistema o identifica como sendo outro [Fiala, 2005].

ARToolKit e o ARTag utilizam padrões para identificação dos marcadores, porém ARTag compara códigos digitais compostos de 0´s e 1´s, ao invés de imagens como o ARTookit, diminuindo o processamento requerido para sua identificação. Outra característica do ARTag é que ele detecta a presença de oclusão e controle de luz, que são pontos falhos do ARToolkit, além de apresentar menor *jitter* da posição detectada em caso de perda de foco. O ARTag está disponível para Windows e Linux, sem código fonte. Disponível em http://www.artag.net.

6.4.4 Dart

Designers Augmented Reality Toolkit (DART) é um editor gráfico. Ele foi desenvolvido como um conjuto de extensões do ambiente de programação multimídia Macromedia Director. Ele é composto por *behaviors* (extensões do Director escritas na linguagem LINGO) e Xtras (*plugins* para o Director escritos na linguagem

C++). Esta ferramenta utiliza o ARToolKit para a captura de vídeo, *tracking* e para o processo de reconhecimento de marcadores [MacIntyre, 2004].

Este projeto é voltado para aplicações onde a mídia gerada por computador é diretamente integrada à percepção dos participantes. Ele suporta Windows e MacOSX. Apesar de estar disponível para uso livremente, ele exige o Macromedia Director, que é um *software* comercial. Disponível em http://www.gvu.gatech.edu/dart/.

	Tipo	Documentação	Fácil de usar	Recursos	Custos	Linguagem de Programação
ARToolkit	API	++	++	++	+++	C
ARToolkit Plus	API	++	++	+++	+++	C++
ARTag	API	+	++	++	+	C++, C#
DART	GUI	++	+++	++	+	Não oferece

Tabela 6.2 - Tabela de comparação de ferramentas de RA.

6.5 Conclusão

O desenvolvimento das ferramentas de RV e RA tem sido contínuo, e estas tendem a se tornar cada vez mais sofisticadas e completas. Estão sendo constantemente melhoradas graças ao desenvolvimento de *hardwares* gráficos cada vez mais sofisticados, que têm melhorado a qualidade e desempenho de geração das imagens. As ferramentas de RV têm adquirido maior realismo gráfico, suporte a formas mais avançadas de animação e programabilidade por meio de *scripts*. Já as ferramentas de RA têm adquirido maior velocidade de processamento e maior estabilidade no reconhecimento de objetos. Da mesma forma, os processos de desenvolvimento de *software* e os *kits* têm também evoluído muito, restando ainda, no entanto, muitos problemas ainda em aberto e

que devem ser pesquisados. Espera-se que os leitores desenvolvam aplicações com estas ferramentas, contribuindo assim com os testes e aprimoramento das soluções de *software* disponíveis.

6.6. Referências Bibliográficas

Alves, W.P. “Modelagem e Animação com Blender”. Editora Érica. São Paulo. 2006.

Brito, A. “Blender 3D:Guia do Usuário”. Editora Novatec. São Paulo. 2006.

Farias, T.S.M.C. and Pessoa, S.A and Teichrieb, V. and Kelner, J. “O Engine Gráfico OGRE”, In: SRV 2006 - VIII Symposium on Virtual Reality. Belém. Pará.

Fiala,M."ARTag, a fiducial marker system using digital techniques".Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02.Pages: 590 – 596.

Guimarães, M. P. Um ambiente de desenvolvimento de aplicações de Realidade Virtual baseadas em aglomerados gráficos. Tese de doutorado. Escola Politécnica da Universidade de São Paulo. Dezembro. 2004.

MacIntyre, B. , Gandy, M., Dow, S., Bolter, J.D. "Tools: DART: a toolkit for rapid design exploration of augmented reality experiences". Proceedings of the 17th annual ACM symposium on User interface software and technology UIST '04. Publisher: ACM Press.2004.

Netto, J. C. M.; Moraes, R. M.; Machado, L. S. “Um Estudo Comparativo de Ferramentas para a Criação de Jogos Educacionais Baseados em Realidade Virtual”. Anais do II Workshop de Aplicações de Realidade Virtual (WARV'2006). Novembro, Recife, Brasil.2006.

Rocha, E.J.T.S. “Forge 16V: Um Framework para Desenvolvimento de Jogos Isométricos”. Dissertação de mestrado.Universidade Federal de Pernambuco. Recife. Dezembro. 2003.

Silva, R.J.M.; Wagner, G.N.; Raposo, A.B.; Gattass, M. “Experiência de Portais em Ambientes Arquitetônicos Virtuais".SVR 2003 - VI Symposium on Virtual Reality, Ribeirão Preto, SP, Brasil, Out. 2003, pp.117-128.

Capítulo 7

Sistemas Distribuídos de Realidade Virtual e Aumentada

Ildeberto Aparecido Rodello[1], Antonio Carlos Sementille[1,2], José Remo Ferreira Brega[1,2] e Fátima L. S. Nunes[1]

[1] Programa de Pós-Graduação em Ciência da Computação (PPGCC) - Centro Universitário Eurípides de Marilia (UNIVEM)
Caixa Postal 2041 – 17525-901 – Marília – SP – Brasil

[2] Laboratório de Sistemas de Tempo Real (LSTR) - Departamento de Computação (Dco) - Universidade Estadual Paulista - Unesp
Caixa Postal 473 - 17.033-360 - Bauru - Brasil

{rodello,semente, remo, fatima}@univem.edu.br

Abstract

This chapter aims at presenting the main concepts related to the design and implementation of Distributed Virtual and Augmented Reality Systems. Related to the design, the factors that should be considered in the definition of the environment are presented. The modalities of implementation are discussed, as well as, the factors that could have impact on the perfect running. About implementation, are highlighted the models of communication, the involved protocols, the ways of implementation and some used indicatives for the performance evaluation.

Resumo

Este Capítulo tem como objetivo apresentar os principais conceitos relacionados ao projeto e implementação de Sistemas Distribuídos de Realidade Virtual e Aumentada. Relacionado ao projeto, são apresentados fatores que devem ser considerados quando da definição do ambiente. São discutidas as possíveis modalidades de implementação, bem como os fatores que podem ter impacto no perfeito funcionamento. Sobre a implementação são destacados os modelos de comunicação, os protocolos envolvidos, as formas de implementação e os indicativos que podem ser utilizados para avaliar o desempenho.

7.1. Introdução

Em virtude da expansão e do desenvolvimento tecnológico das redes de comunicação, a concepção e implementação de sistemas que têm o aspecto distribuído e/ou compartilhado como principal agente motivador tem crescido. Atualmente é possível utilizar-se da "rede" para diversas atividades, incluindo negócios, entretenimento e disseminação de conhecimento, entre outros.

Dentro desse contexto, a utilização da Realidade Virtual (RV) e da Realidade Aumentada (RA) de forma distribuída tem exigido estudos e discussões a respeito dos fatores que podem ser levados em consideração para atingir o sucesso no desenvolvimento de tais tipos de ambiente. É interessante destacar que tais fatores devem ser considerados tanto para o uso dos modelos de comunicação comuns (face às aplicações cotidianas), quanto para aspectos específicos de ambientes de RV e/ou RA.

Assim, o presente Capítulo procura apresentar e discutir esses fatores, disponibilizando ao leitor a base necessária para a tomada de decisões no que diz respeito ao projeto e a implementação de Sistemas Distribuídos de Realidade Virtual (SDRV) e Aumentada.

7.2. Projeto

O projeto de um SDRV envolve uma série de requisitos que compreendem principalmente a modalidade adotada e os cuidados com uma série de fatores que afetam seu desempenho. Essa seção apresenta e discute tais requisitos.

7.2.1 Modalidades

Considerando-se a finalidade do ambiente virtual (AV) é possível classificar um SDRV e Aumentada em: multiusuário, colaborativo, de grande escala ou de realidade aumentada.

Ambientes Virtuais Multiusuário

Os AVs multiusuário compreendem aqueles AVs que simplesmente utilizam a infra-estrutura de uma rede de computadores para simular a interação em tempo real entre vários usuários em um mundo virtual tridimensional compartilhado.

Nessa categoria considera-se que não há qualquer ação de auxílio entre os participantes. Cada usuário é representado no AV compartilhado por uma entidade (avatar) que é visível, possivelmente sob outro ponto de vista, no computador de cada outro participante.

Ambientes Virtuais Colaborativos

Em AVDs os participantes podem compartilhar um mesmo espaço virtual de trabalho (*workspace*), onde poderão ajudar-se na execução de uma determinada tarefa, usando os princípios de trabalho cooperativo baseado em computador (*CSCW – Computer Supported Cooperative Work*). Nesse sentido, classificar-se-á o sistema como um Ambiente Virtual Colaborativo (AVC) [Benford et al., 1994].

Observa-se que o principal diferencial de um AVC é a possibilidade de cooperação entre os usuários na execução de uma determinada tarefa. Em resumo, as propriedades desse ambiente relacionam-se com: espaço, presença e tempo compartilhados; comunicação entre os participantes e; interação com o ambiente [Snowdon et al., 2001].

Ambientes Virtual de Grande Escala

Os Ambientes Virtuais de Grande Escala (AVGEs) caracterizam aqueles ambientes que ganham grandes proporções no que diz respeito ao seu tamanho, quantidade de objetos presentes, quantidade de participantes e complexidade de gerenciamento, tanto que, por questões de desempenho, o AV pode ser dividido e o controle sobre estas partes ser exercido por diferentes máquinas.

Durante a navegação pelo ambiente, o usuário pode penetrar em partições fora do seu domínio, de forma que sua máquina ou servidor deverá receber uma réplica dessa partição. Se existirem vários participantes em uma mesma região do AV, esse grupo de participantes receberá uma cópia dessa região e, qualquer alteração no AV feita por um membro do grupo, será retransmitida somente para o restante do grupo.

Ambientes Virtuais Distribuídos de Realidade Aumentada

A interface para aplicações de RA ainda não tem um grau de desenvolvimento tal como as tradicionais de RV. A inserção de objetos virtuais no ambiente real requer diferentes formas de interação, em adição àquelas do ambiente gerado computacionalmente.

Da mesma forma, os aspectos inerentes à distribuição também necessitam de cuidados especiais, uma vez que deve haver uma maneira de compartilhar o mundo real entre os participantes.

Assim, os requisitos necessários para ambientes de RA distribuídos têm, adicionalmente, a preocupação com transmissão de imagens do mundo real em tempo real. Assim, os requisitos necessários para a transmissão de informações são diferentes, exigindo principalmente, mais largura de banda e qualidade de serviço.

7.2.2. Fatores de influência

Assim como qualquer outro AV, a criação de AVDs está bastante ligada com realismo visual e interação. Dessa forma, assuntos como modelagem gráfica tridimensional e interação homem-máquina constituem uma parte fundamental. Em especial destaca-se o suporte de comunicação em rede. Por meio de uma rede é fornecida a base para que unidades computacionais separadas fisicamente sejam unificadas para implementar um único AV.

Além disso, os fatores adiante apresentados e comentados afetam, de alguma forma, o desempenho e/ou a complexidade de um AVD. A complexidade, por sua vez, também afeta o desempenho, prejudicando o realismo, na forma de atrasos ou lentidão.

Dessa forma, procura-se estabelecer um nível de complexidade que torne o AVD aceitável, tanto em realismo quanto em desempenho. Os principais fatores que devem ser levados em consideração para que isso ocorra são [Greenhalgh, 1996b]:

Número de participantes

De acordo com a quantidade de participantes que o AV pode suportar de forma simultânea, os sistemas de RV podem ser classificados em monousuário (*single user*) ou multiusuários (*multi-user*).

Em um sistema monousuário é possível a existência de outros usuários que são, entretanto, somente meros espectadores, uma vez que a interação com e a navegação pelo ambiente somente são disponíveis a um único usuário.

Nota-se que os sistemas multiusuários são mais complexos e exigem um poder maior de computação, relacionando-se, entre outros, a aspectos como gerenciamento desses usuários (*login*, segurança, etc.). Além disso, o número de tarefas que poderão ser executadas pelos usuários é outro fator de interferência.

Percepção

Em um AVD é de fundamental importância que os participantes tenham a percepção da presença de outros participantes e das ações que estes executam sobre o AV.

Quando um participante entra em um local compartilhado, este se torna uma "pessoa virtual", denominado de *avatar*, ou seja, assume uma representação gráfica dentro do ambiente. Uma vez dentro de um AVD, cada participante pode visualizar outros avatares localizados no mesmo espaço e suas interações no ambiente. Similarmente, quando um participante deixa o AVD, os outros participantes vêem seu *avatar* partir. Nem todos os avatares precisam ser controlados por participantes. Alguns podem ser entidades sintéticas controladas por modelos de simulação dirigidos por eventos.

Quantidade de Objetos Presentes no AV

Os objetos que compõem cada AV também são fatores a serem discutidos em termos de quantidade, nível de detalhes e comportamento. Eles estão relacionados de forma que um equilíbrio contribui para o desempenho do sistema. A quantidade de objetos corresponde a quantos objetos estarão presentes na cena,

de acordo com certo nível de detalhamento e um comportamento permitido.

Objetos mais próximos necessitam de um nível de detalhes maiores, implicando em um número maior de polígonos. Tal nível, entretanto, não deve ser mantido para o mesmo objeto, quando a uma distância maior, uma vez que o realismo tende a ser mantido, reduzindo-se os detalhes dos objetos à medida que o ponto de vista deles se afasta.

O comportamento de um objeto depende principalmente do nível de interação que o usuário pode ter. Objetos que o usuário pode somente observar têm um nível de complexidade menor que aquele com o qual o usuário pode interagir, modificar e ajustar.

O padrão de comportamento, entretanto, depende da finalidade do mundo virtual. AVCs necessitam de um padrão comportamental mais apurado, principalmente se estiver sendo utilizado para treinamento [Oliveira et al., 2000].

Em um AVC, o usuário pode ainda ter a possibilidade de exploração do objeto segundo dois aspectos: *objetivo* ou *subjetivo* [Greenhalgh, 1996a]. No aspecto objetivo, os usuários estão em consenso a respeito de um determinado objeto, todos interessados nas mesmas informações. No aspecto subjetivo, o usuário poder ter acesso a informações exclusivas e adicionais a respeito do objeto, isto é, além daquelas presentes no aspecto objetivo.

O modelo subjetivo leva o usuário a conseguir uma maior interação com o ambiente, de forma que este passa a oferecer um conjunto maior de operações disponíveis. Dentre as operações pode-se citar o acesso à base de dados exclusivos de onde o usuário possa filtrar informações.

Particionamento

Pelo fato de os seres humanos serem orientados visualmente,

espera-se ver a representação gráfica do AV com todos os detalhes. Porém, o computador não precisa necessariamente *renderizar* o mundo virtual completo, uma vez que a capacidade visual humana não permite receber um grande nível de detalhe de objetos distantes e ocultos. Somente os objetos ao redor de uma determinada região onde se encontra o participante, precisam ser *renderizados* com um grande nível de detalhe.

Em alguns casos, como na detecção de colisão, pode-se necessitar de taxas de atualizações que se aproximem das taxas de quadros executados localmente, de maneira a alcançar um maior realismo.

Assim, uma solução viável é dividir o AV em áreas menores, melhor gerenciáveis, de acordo com o interesse do usuário, aumentando a escalabilidade e o processamento das informações.

Segundo Macedonia et al. (1995), é possível particionar o AVD em *grids* (retângulos) ou em hexágonos. A associação baseada em *grids* tem a desvantagem de ter muitos pontos em que vários retângulos se encontram, apesar de ser mais fácil a implementação. Quando um *avatar* se aproxima de qualquer um desses pontos, o *host* precisa renderizar (como forma de antecipação das células) os quatro retângulos próximos à intersecção. Para diminuir esta dificuldade de processamento, pode-se dividir o AV em hexágonos, nos quais somente três regiões se cruzam em um determinado ponto, ao invés de quatro.

7.3. Questões de Implementação

Consciente das questões relacionadas ao projeto é necessário tomar decisões de como implementá-las. Nessa seção são abordados aspectos relativos aos modelos de comunicação que podem ser adotados, aos protocolos que podem ser utilizados, aos modelos de suporte a implementação e, por fim, os indicativos para avaliação de desempenho.

7.3.1. Modelos de Comunicação

Esses tópicos referem-se às formas como as máquinas se organizam para suportar a troca de mensagens entre os AVs.

Cliente/Servidor

O paradigma Cliente/Servidor em um AVD implica na comunicação direta somente do cliente com o servidor ou vice-versa, não havendo a troca de mensagens entre os clientes. É o modelo mais simples. As vantagens e desvantagens deste modelo estão associadas aos fatores básicos de qualquer modelo Cliente/Servidor.

Uma desvantagem é a possibilidade de falha no servidor, o que tornaria o sistema inoperante. Para contornar esse problema, a técnica da replicação de servidores pode ser utilizada. Além disso, o servidor é um gargalo em potencial e, em se tratando de um AVD onde coexistem vários usuários, danos podem ser causados em termos de latência. Um eficiente mecanismo de balanceamento de cargas pode solucionar esse problema.

Peer to Peer

O modelo *Peer-to-Peer* implica na comunicação direta entre os *hosts*, sem intermédio de um ou mais servidores. O ponto de estudos neste modelo refere-se à complexidade de gerenciamento do ambiente, uma vez que além do gerenciamento local, cada *host* deve também tratar os acessos remotos.

7.3.2. Protocolos

Segundo Kurose e Ross (2006) um protocolo "*define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou*

no recebimento de uma mensagem ou outro evento”. Por envolver comunicação, o protocolo é um importante componente de um SDRV e Aumentada.

Essa seção apresenta uma visão geral do protocolo TCP/IP, que é, de maneira geral, o protocolo mais utilizado na transmissão de informações. Apresenta também alguns protocolos específicos para SDRVs e, por fim, discute superficialmente dois protocolos para redes sem fio (*wireless*), visto que a mobilidade se torna cada vez evidente no cotidiano.

TCP/IP

A arquitetura TCP (*Transmission Control Protocol*) / IP (*Internet Protocol*) representa um grupo de protocolos que funcionam em conjunto para transmitir informações pela Internet. Assim, não se usa somente um protocolo, combinam-se protocolos de diferentes camadas para diferentes propósitos. O TCP e o IP são apenas dois dos protocolos de transmissão abrangidos pela arquitetura TCP/IP.

O *Internet Protocol* (IP) é o protocolo que define o mecanismo de transmissão não confiável e não orientado à conexão. O IP define a unidade básica de transferência e o formato exato de todos os dados que passam através da Internet. Ele escolhe o caminho pelo qual serão enviados os pacotes ou datagramas (roteamento) do *host* origem ao *host* destino [Comer, 1998]. Os *hosts* são identificados por endereços de tamanho fixo, chamados endereços IPs. Apesar dos esforços para entregar o datagrama no destino, o IP não garante essa entrega. A garantia de entrega, assim como o controle de ordenação do fluxo de dados é de responsabilidade dos protocolos de camada mais alta, como o protocolo TCP da camada de transporte [Albuquerque, 2001].

O *User Datagrama Protocol* (UDP), em contraste com o TCP, não garante que as mensagens serão entregues na ordem

correta nem se elas serão recebidas. As mensagens são limitadas ao tamanho de um único pacote (datagrama), que utiliza a transmissão de dados pela rede sem conexão, ou seja, o envio de mensagens consecutivas sem a espera de uma confirmação do destinatário. Desse modo, o UDP não é um protocolo confiável e para certas aplicações é necessária a retransmissão de mensagem.

Apesar dessa desvantagem, o UDP é muito mais rápido que o TCP e pode ser usado para aplicações em que os pacotes perdidos não tenham tanta importância como na transmissão de áudio e vídeo [Diehl, 2001].

DIS

Em 1985, o *Departament of Defense* dos Estados Unidos (DoD) iniciou um projeto chamado SIMNET [Macedonia, 1996] com o objetivo de desenvolver um sistema de treinamento de guerra com tanques, aeronaves, projéteis e outros que suportassem muitos computadores simultâneos em rede.

Após o SIMNET, iniciou-se o projeto *Distributed Interactive Simulation* (DIS) em 1989, de onde surgiu um protocolo padrão com o mesmo nome, reconhecido IEEE em 1993. Baseado em experiências com o DIS, surgiu o projeto *High-Level Architecture* (HLA) [Dahmann et al., 1997] utilizado para aplicações distribuídas juntamente com o *Common Object Request Broker Architecture* (CORBA) [Corba, 2001].

Apesar de sua aplicação restrita, o DIS introduziu idéias que influenciaram o surgimento de outros protocolos. A especificação do DIS contém 27 formatos de mensagem PDU. Há PDUs para cada serviço específico, por exemplo, detonação, disparos de projéteis ou marcação de campos minados. A PDU mais importante é a de estado da entidade, utilizada para transmitir o estado completo do objeto, posição, orientação e velocidade.

O DIS exige que cada entidade envie constantemente mensagens de estado dos objetos que estão ativos, para todos os *hosts*. Ele é um protocolo *peer-to-peer*, mas devido às transmissões constantes das entidades ativas, ele não suporta grande número de usuários e, assim, não pode melhorar sua escalabilidade [Diehl, 2001].

VRTP

O *Virtual Reality Transfer Protocol* (VRTP) foi desenvolvido pela DARPA, e espera-se que tenha o mesmo papel em AVD para o VRML (*Virtual Reality Modeling Language)*, quanto o HTTP (*Hyper Text Transfer Protocol*) tem para as páginas HTML (*Hypertext Markup Language)*. São necessários outros recursos além do HTTP na realização dessa tarefa, que combinem a comunicação *peer-to-peer*, cliente/servidor e monitoramento da rede [Brutzman et al., 1997].

Um computador consegue executar várias tarefas ao mesmo tempo no VRTP. Ele pode ser um cliente, um *host* ou um *peer*, isto é, um participante com direitos iguais. A tarefa de um cliente é representar uma cena, por exemplo, no navegador VRML. Um servidor gerencia uma cena e envia para os clientes, por exemplo, pelo HTTP. A parte crucial da arquitetura do VRTP é a comunicação *peer-to-peer* e o *multicast.*

DWTP

O *Distributed Worlds Transfer and Communication Protocol* (DWTP) foi desenvolvido na *German National Research Center for Information Tecnology* (GMD) para a camada de aplicação de AVGEs. Em contraste com o DIS, o DWTP é heterogêneo e independente da aplicação, isto é, dados de diferentes tipos podem ser transmitidos.

Esses protocolos utilizam mensagens de reconhecimento negativo (NACK – *Negative Acknowledgement*) em uma

comunicação fim-a-fim, ou seja, precisa ser confirmado o não recebimento correto da mensagem. Os erros na transmissão podem ser causados por roteadores *multicast*, e estes roteadores são sobrecarregados, devido aos NACKs adicionais de cada cliente. Alguns protocolos tentam evitar isso, através de um tempo de espera antes do envio de um *NACK*; similar ao usado no protocolo *Ethernet* para resolver os conflitos de barramento.

No DWTP, um número pequeno de destinos envia o reconhecimento de mensagem contendo o número de todos os participantes que estão registrados. Assim, um participante pode detectar o problema ao receber um *ACK* de um pacote e não o próprio pacote. Nesse caso, o pacote é retransmitido por *unicast* confiável (TCP) para o participante que o solicitou [Diehl, 2001].

Multi-user 3D Protocol

O protocolo *Multi-user 3D Protocol* (Mu3D) foi desenvolvido por Galli e Luo na *University of Balearic Islands,* para um projeto de arquitetura colaborativa. Em contraste com o DWTP, o protocolo Mu3D é *peer-to-peer*; já em contrate com o DIS, ele não envia os dados completos, somente as atualizações [Diehl, 2001].

Suponha-se, por exemplo, um jogo de tabuleiro distribuído em que só possa haver um objeto em um quadrado por vez. Suponha que a ordenação da origem foi violada: o cliente 2 não recebe a mensagem na ordem em que foi enviada pelo cliente 1. Pode-se forçar a ordenação na origem ao se usar números seqüenciais. Se um cliente receber uma mensagem com um número de seqüência maior que o último, ele espera até que tenha recebido todas as mensagens intermediárias.

O cliente bloqueia o nó, enquanto estiver fazendo alterações, e libera somente depois de enviar a mensagem da alteração para todos os clientes. Devido ao fato do Mu3D enviar atualizações e não o estado completo da entidade, o tamanho das mensagens,

incluindo todos os *overheads* das camadas inferiores do protocolo, é pequena (200 a 300 *bytes*) [Diehl, 2001].

IEEE 802.11

O padrão 802.11 apareceu na década de 90, mais especificamente em 1997, tornando-se uma das tecnologias principais no mundo *wireless*. Este padrão define as camadas de Controle de Acesso ao Meio (MAC) e física (PHY) para uma LAN com conectividade wireless.

A arquitetura 802.11 compreende diversos componentes e serviços que se interagem para fornecer, mobilidade e transparência às camadas mais elevadas da pilha de protocolos da rede. Basicamente é composto por [Kurose e Ross, 2006]:

802.11: Usando FHSS (*Frequency Hopping Spread Spectrum*) ou DSSS (*Direct Sequence Spread Spectrum*), fornece taxa de transmissão de 1 a 2 Mbps, operando na faixa de freqüência de 2.4GHz.

802.11a: Usando OFDM (*Orthogonal Frequency Division Multiplexing*) fornece taxa de transmissão de até 54Mbps e opera na faixa de freqüência de 5GHz.

802.11b: Também conhecido como *Wi-Fi* ou a *High Rate* 802.11, usa DSSS e aplica-se a LANs wireless. É geralmente utilizado para uso confidencial, em residências. Fornece uma taxa de transmissão de 11 Mbps.

802.11g: Fornece uma taxa da transmissão de 20+ Mbps e aplica-se a LANs. Funciona na faixa de 2.4GHz.

O padrão 802.11 pode assumir ainda diversas topologias de interconexão, que compreendem: *Independent Basic Service Set (IBSS)* ou *Ad-Hoc*, *Basic Service Set (BSS)* e *Extended Service Set (ESS).*

BlueTooth

Bluetooth é um tipo simples de rede *wireless* que permite a formação de uma pequena rede com os até oito dispositivos que conectados de uma vez. Tais dispositivos incluem PDAs, Laptops, telefones móveis e computadores pessoais. Entretanto, o Bluetooth pode também ser encontrado nos teclados, mouses, *headsets* e jogos *hands-free* de telefone móvel, entre outros. Foi inventado originalmente por Ericsson em 1994.

Dentre suas principais vantagens é possível destacar o baixo custo e a configuração dinâmica. É interessante citar que o Bluetooth, ao contrário de outros padrões, compreende não apenas as camadas mais baixas da rede, mas também a camada da aplicação.

7.3.3. Formas de Implementação

São vários os modelos de suporte de comunicação que podem ser utilizados. Geralmente o desenvolvedor procura um ambiente que lhe abstraia detalhes de comunicação. São vários os pacotes de software que oferecem tais serviços. Pode-se dividi-los em quatro categorias: baseados em *sockets, toolkits, middlewares* e *frameworks*.

Os *sockets* são primitivas que estabelecem um canal de comunicação inter-processos, por meio de chamadas de sistemas. No *socket* estabelece-se uma porta de comunicação onde mensagens são enviadas e recebidas, baseando-se em um domínio de endereço (*Unix Domain ou Internet Domain*) e em um tipo (*Stream Sockets ou Datagram Sockets*). Como exemplo pode-se citar aplicações desenvolvidas na linguagem Java e VRML [Kubo, 2000; Kirner et al., 2001].

Na categoria *toolkits* se enquadram suporte os modelos desenvolvidos especificamente para um conjunto de aplicações. Como exemplo pode-se citar o *World2World* (W2W) e *WorldToolKit* (WTK) [Sense8, 1998], DIVE [Hagsand, 1996], MASSIVE [Greenhalgh; Benford, 1995], entre outros.

O *World2World*, por exemplo, é uma solução em rede baseada no modelo cliente/servidor para simulações 3D interativas em tempo real. Os componentes clientes do *World2World* integram-se tanto com *WorldToolKit* quanto *WorldUp* [Sense8, 1998], facilitando o desenvolvimento de simulações multiusuários em ambos os produtos.

Os *middlewares* são soluções independentes de plataformas e de domínio de aplicação. Como exemplo pode-se citar o modelo CORBA (*Common Object Request Broker Architecture*) [Corba, 2001], Java RMI (*Remote Method Invocation*) [Javarmi, 2002] além de outros.

Por fim, têm-se os *frameworks*, que são soluções específicas para um determinado domínio de aplicação.

Nesta linha, o AVOCADO, atualmente denominado AVANGO apresenta um *framework* orientado a objetos para o desenvolvimento de ambientes virtuais interativos e distribuídos [Tramberend, 1999]. Ele foi baseado no SGI Performer, tendo as linguagens de programação C++ e Scheme como plataformas para desenvolvimento.

Uma abordagem que está sendo cada vez mais usada segue somente a adoção da plataforma Java. Nesse modelo pretende-se tirar proveito do paradigma Java "*write once, run* anywhere", aliado a seu baixo custo. O *framework* DISCIPLE [Krebs, 2003] é um exemplo que procura prover colaboração em redes heterogêneas (como a web), suportando inclusive, computação móvel.

Existem ainda trabalhos como o *Extensible Modeling and Simulation Framework (XMSF)*, que tem por objetivo definir um conjunto de tecnologias baseadas na web, dentro de um *framework* extensível, para permitir a criação de uma nova geração de aplicações de modelagem e simulação, especificamente aquelas apoiadas em XML (*Extensible Markup Language*) [Xsmf, 2003].

A utilização de soluções proprietárias tem o custo como um fator negativo. As aplicações conseguem alcançar bons índices de atrasos e escalabilidade, porém tem-se alto custo com licenças.

Segundo Deriggi (1998) e [Deriggi, et al., 1999], os sistemas desenvolvidos com base em *toolkits* tendem a ter um desempenho melhor em comparação com outras plataformas, entretanto apresentam problemas relacionados a portabilidade e dependência de plataforma específica. Propostas têm sido apresentadas como em Deriggi et al. (1999), Sementille [Sementille, 1999] e Kubo [Kubo, 2000]. A desvantagem associada ao desenvolvimento de AVD com *sockets* refere-se principalmente a escalabilidade.

7.3.4. Indicativos para Avaliação de Desempenho

A grande maioria dos aspectos discutidos até o momento, afeta de alguma forma o desempenho do AVD, podendo então ser analisados segundo o prisma de confiabilidade e atraso.

A confiabilidade da rede é uma medida que reflete a quantidade de dados que são perdidos pela rede durante o trajeto do *host* origem para o *host* destino. O dado perdido pode ser classificado como: dado interrompido e dado corrompido.

O dado interrompido é quando o dado não chega por completo ao *host* destino. O dado corrompido é quando o conteúdo dos pacotes de dados é alterado durante a transmissão e o que chega ao *host* destino é basicamente inútil. Muitos pesquisadores consideram a interrupção e a corrupção de dados essencialmente

equivalentes, isto porque, em ambos os casos, o dado não atinge efetivamente a aplicação destino. A confiabilidade da rede freqüentemente força um compromisso entre a largura de banda e a latência [Singhal e Zyda, 1999].

Em geral, o atraso é um dos principais fatores limitantes para aplicações de Realidade Virtual. Segundo Wloka (1995) as principais fontes de atraso encontradas em sistemas de RV são: atraso de dispositivos de entrada do usuário, atraso do processamento, atraso de sintetização, atraso de sincronização e atraso induzido pela taxa de quadros por segundo.

Além desses fatores, a escalabilidade [Tanenbaum, 1995] é um outro ponto a ser analisado. Ela consiste em se analisar a capacidade de crescimento de um sistema em termos de usuários conectados ao mundo. Quanto maior a quantidade de usuários, maiores serão as dificuldades de gerenciamento do mundo e por conseqüência, o sistema fica sujeito a atrasos empobrecendo o desempenho.

7.4. Ambientes Virtuais Distribuídos de RV e RA e os Dispositivos Móveis

Com o advento e a propagação das redes *wireless* e dos dispositivos móveis, várias aplicações de RV e, principalmente RA, têm surgido com suporte dessa tecnologia.

Segundo Wagner e Schmalstieg (2003) a RA é um complemento natural à computação móvel, uma vez que sistemas de RA móveis podem ajudar o usuário em diversas situações.

Nesse sentido, os autores destacam diversos projetos como o MARS (*Mobile Augmented Reality Systems*) [Höllerer et al., 1999] que permite o usuário caminhar livremente, com todo o equipamento colocado nas suas costas e o Bat System [Newman at al., 2001] pertencente a AT&T.

7.5. Considerações Finais

O objetivo final de um AVD é reunir usuários geograficamente dispersos e permitir que eles tenham capacidade de utilização do sistema. Nesse aspecto os fatores aqui apresentados e discutidos influem diretamente, de forma que o desenvolvedor deve estar atento a eles.

Sugere-se uma análise com o intuito de equilibrá-los, construindo um sistema com um bom grau de realismo e que tenha desempenho aceitável. Como cada vez mais a heterogeneidade dos sistemas se torna um fator destacável, sugere-se que o modelo de suporte para comunicação abranja a maior quantidade possível de arquiteturas.

Destaca-se ainda que, procurou-se apresentar os aspectos mais gerais, não se relacionando a um modelo específico, que pode ter suas peculiaridades e aspectos específicos. A Bibliografia é recomendada aos que desejarem se aprofundar nos assuntos aqui tratados.

7.6. Referências Bibliográficas

Albuquerque, F. (2001) TCP/IP Internet: Programação de Sistemas Dstribuídos – HTML, Javascript e Java. Rio de Janeiro: Axcel Books do Brasil Editora.

Benford, J. B., Fahlén, L. E., Mariani, J., Rodden, T. (1994) Supporting Cooperative Work in Virtual Environments, The Computer Journal, V. 37, N. 8, pp. 653-667.

Brutzman, D.; Zyda, M.; Watsen K.; Macedonia, M. (1997) Virtual reality transfer protocol (VRTP) design rationale. In: Proc. f the IEEE Sixth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise. (WETICE'97). IEEE Computer Society. p. 179-186.

Comer, D. E. (1998) Interligação em rede com TCP/IP: Princípios, protocolos e arquitetura. Rio de Janeiro.Editora Campus.

Corba. (2001). Common Object Request Broker Architecture. Disponível em <http://www.corba.org>.

Dahmann, J. S.; Fujimoto, R. M.; Weatherly, R. M. (1997) The department of defese high level architecture. In: Winter Simulation Conference. Disponível em: <http://www.cc.gatech.edu/computing/pads/PAPERS/DOD_High_Level_Arch.pdf>. Acesso em: 04 mar. 2003.

Deriggi Jr., F. V. (1998) Suporte de Comunicação para Sistemas Distribuídos de Realidade Virtual, Dissertação (Mestrado), UFSCar, Agosto.

Deriggi Jr., F., Kubo, M.M., Sementille, A.C., Santos, S.G., Brega, J.R.F., Kirner, C. (1999) CORBA Platform as Support for Distributed Virtual Environment. In: Proc. of the IEEE VR.

Diehl, S. (2001) Distributed Virtual Worlds: Foundations and Implementation Techniques Using VRML, Java, and Corba. Verlag; Berlin; Heidelberg; Springer.

Greenhalgh, C. (1996a) Approaches to Distributed Virtual Reality Systems. Relatório Técnico (NOTTCS-TR-96-5); Universidade de Nottingham.

Greenhalgh, C. (1996b) Supporting Complexity in Distributed Virtual Reality Systems. Relatório Técnico (NOTTCS-TR-96-6); Universidade de Nottingham.

Greenhalgh, C., Benford, S. (1995) MASSIVE: A Collaborate Virtual Environment for Teleconferencing. ACM Transactions on Computer-Human Interaction, V. 2, N. 3, pp. 239-261.

Hagsand, O. (1996) Interactive Multiuser Ves in the DIVE System, IEEE Multimedia, Spring, V. 3, N. 1, pp. 30-39.

Höllerer T., Feiner S., Terauchi T., Rashid G., Hallaway D., (1999) Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System, Computers and Graphics, 23(6), Elsevier Publishers, pp. 779-785.

JavaRMI. (2002) Getting Started using RMI . Disponível em: <http://java.sun.com/j2se/1.4/docs/guide/rmi/getstart.doc.html>.

Kirner T.G., Kirner C., Kawamoto A.L.S., Cantão J., Pinto A. Wazlawick R.S. (2001) Development of a Collaborative Virtual Environment for Educational Applications. In: Proceedings of the ACM Web3D 2001 Symposium, pp. 61-67.

Kurose, J. F., Ross, K. W., (2006) Redes de Computadores e a Internet. Addison Wesley.

Krebs, A. M., Ionescu, M., Dorohomceanu, B., Marsic, I. (2003) The DISCIPLE System for Collaboration over the Heterogeneous Web. In: Proceedings of the Hawaii International Conference on System Sciences.

Kubo, M.M. (2000) Suporte de Comunicação para Sistemas Colaborativos Interoperáveis de Realidade Virtual. Dissertação (Mestrado). UFSCar. São Carlos. SP.

Macedonia, M. R., Zyda, M. J. (1996) A Taxonomy for Networked Virtual Environments. IEEE Multimedia. V. 4, N. 1. pp. 48-56.

Macedonia, M. R., Zyda, M. J., Pratt, D. R., Brutzman, D. P., Barham, P. T. (1995) Exploiting Reality With Multicast Groups. IEEE Computer Graphics and Applications. V. 15. N. 9.

Newman J., Ingram D., Hopper A., (2001). Augmented Reality in a Wide Area Sentient Environment. Proceedings of the 2nd International Symposium on Augmented Reality (ISAR 2001).

Oliveira, J. C.;.Shen X, Georganas N.D. (2000) Collaborative Virtual Environment for Industrial Training and e-Commerce. In: Workshop on Application of Virtual Reality Technologies

for Future Telecommunication Systems. IEEE Globecom'2000 Conference.

Sementille, A. C. (1999) A Utilização da Arquitetura CORBA na Construção de Ambientes Virtuais Distribuídos, Tese (Doutorado) - Instituto de Física de São Carlos - USP, São Carlos, 186p.

Sense8 Corporation. (1998). WorldToolKit Reference Manual – Release 7. Mill Valley. CA.

Snowdon, D., Churchill, E. F., Munro, A. J. (2001) Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction. Springer Verlag.

Tanenbaum, A. S. (1995) Distributed Operating Systems. Prentice-Hall.

Tramberend, H. (1999) Avocado: A Distributed Virtual Reality Framework. Proceedings of the IEEE Virtual Reality. Houston. Texas.

Wagner, D., Schmalstieg, D. (2003) First Steps Towards Handheld Augmented Reality. Proceedings of the 7th International Conference on Wearable Computers, White Plains, NY, USA.

Wloka, M. M. (1995). Lag in Multiprocessor Virtual Reality. Presence. Vol. 4, Nro.1. pp. 50-63.

Xsmf. (2003). Extensible Modeling and Simulation Framework. Disp. em <http://www.movesinstitute.org/xmsf/xmsf.html>.

Zyda, M.; Singhal, S. (1999) Networked Virtual Environments: Design an Implementation. Addison Wesley Pub.

SEÇÃO C

APLICAÇÕES

Capítulo 8

Dispositivos Hápticos para Interfaces de Realidade Virtual e Aumentada

Liliane dos Santos Machado

Departamento de Informática – Universidade Federal da Paraíba (UFPB)
Cidade Universitária s/n – 58052-900 – João Pessoa – PB – Brasil

liliane@di.ufpb.br

Abstract

This chapter presents the main technologies related to the use of interaction devices for touch and force feedback in virtual and augmented reality systems. Thus, devices and control packages that compose haptic systems are presented, as well as their use in some applications.

Resumo

Este capítulo visa apresentar os conceitos relacionados à utilização de interação com sensação de toque e força em sistemas de realidade virtual e realidade aumentada. Assim, serão abordados aspectos relacionados aos equipamentos e aos pacotes de controle que formam os sistemas hápticos, além de serem apresentadas algumas aplicações que os utilizam.

8.1. O que é háptico?

A exploração do mundo real muitas vezes depende do sentido do tato. Esta necessidade evidencia-se em situações onde a visão e/ou a audição não são suficientes para permitir o reconhecimento de objetos, como quando os objetos ficam ocultos visualmente atrás de outros ou o ambiente está escuro. Em mundos virtuais esta mesma situação pode ocorrer. Neste caso, é necessário o uso de sistemas hápticos acoplados aos sistemas de RV ou RA para que a identificação dos objetos seja possível.

O termo háptico deriva-se da palavra grega *hapthai* (toque) e relaciona-se à informação sensorial recebida através do toque ou contato físico. Sabe-se que o toque é o único sentido humano bidirecional, ou seja, recebe e envia informações: ao realizar um toque, uma força é aplicada sobre algo e sua resultante é percebida por quem a aplicou (Figura 8.1). Assim, devido às características do sistema sensorial tátil, realizar ou perceber um toque depende necessariamente do contato direto com o objeto tocado [Sherman e Craig, 2003].
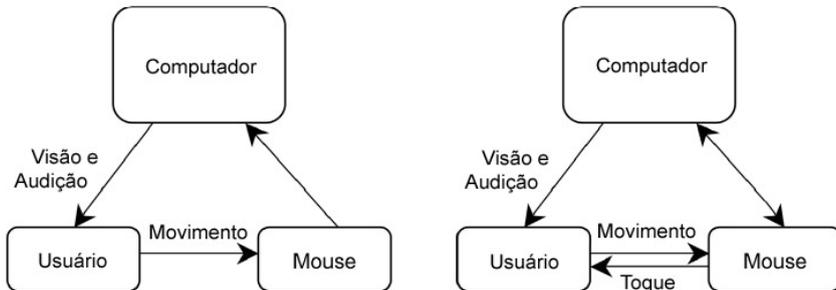
Figura 8.1. Comparação entre o sentido do toque em relação a outros sentidos na comunicação humano-computador.

A percepção do toque está relacionada a duas componentes: tato e cinestesia. O tato permite identificar sensações como

temperatura, pressão ou vibrações, e depende da sensibilidade cutânea. Esta sensibilidade varia de acordo com a região utilizada para realizar o contato. Por exemplo, sabe-se que a ponta dos dedos possui uma elevada capacidade de reconhecer detalhes, superior à palma das mãos [Seow, 1988], devido à quantidade de receptores nela localizados. A cinestesia, por sua vez, refere-se à percepção das tensões aplicadas aos músculos e juntas. Esta percepção é também chamada de propriocepção ou *force feedback* (retorno de força) [Sherman e Craig, 2003] e permite identificar a rigidez de objetos. No ser humano, as sensações de tato e *force-feedback* não podem ser separadas, mas o mesmo não acontece com os equipamentos desenvolvidos para simular estas sensações. Por esta razão há dispositivos que fornecem ambas ou apenas uma delas.

Em sistemas computacionais, a identificação do tátil e de *force-feedback* depende de duas partes igualmente importantes que formam os sistemas hápticos: os dispositivos, responsáveis por receber ações do usuário e apresentar-lhe as propriedades relacionadas ao toque, e as rotinas de controle, responsáveis por calcular e enviar as propriedades do toque ao dispositivo. Inicialmente desenvolvidos para auxiliar pessoas cegas e surdas [Tan e Pentland, 2001], estes sistemas passaram a ser utilizados em aplicações de RV e RA para adicionar realismo. Em geral, são utilizados apenas em partes do mundo virtual, mas seus resultados potencializam a sensação geral de realismo para o usuário.

Na sessão a seguir serão apresentadas algumas das tecnologias utilizadas na criação de dispositivos hápticos, com exemplos de dispositivos encontrados no mercado. Em seguida serão apresentadas aplicações de sistemas hápticos em sistemas de RV e RA.

8.2. Tecnologias

Nos últimos anos, uma série de dispositivos hápticos vêm sendo desenvolvidos com o objetivo de oferecer meios de reconhecer e manipular objetos em sistemas de RV e RA. Estes dispositivos podem ser divididos em três categorias principais, de acordo com seu uso: para as mãos, para os braços ou pernas, e para o corpo. Os dispositivos para as mãos geralmente oferecem apenas retorno tátil e permitem sentir as propriedades térmicas ou da superfície de objetos (vibrações, por exemplo). Entretanto, alguns são adicionados de mecanismos que permitem também simular *force-feedback.* Os dispositivos para braços ou pernas visam permitir sentir as forças de resistência relacionadas ao material que compõe o objeto e oferecem retorno de força, ou seja, se utilizam da tensão de músculos ou juntas para permitir sentir a densidade de um objeto ou do ambiente, por exemplo. Já os dispositivos para o corpo, trabalham com o equilíbrio corporal e com a tensão muscular.

Por se tratarem de equipamentos através dos quais o usuário atua e recebe resposta, os dispositivos hápticos são compostos por sensores e atuadores. Os sensores são responsáveis por capturar e enviar ao computador os dados do movimento que serão tratados pelas rotinas hápticas para calcular as informações de resposta enviadas ao dispositivo através dos atuadores.

8.2.1. Dispositivos para mãos

Os dispositivos para mãos costumam fornecer sensações de tato para os dedos ou para as mãos do usuário, como pressão, calor ou vibrações. Para as sensações de pressão, atuadores pneumáticos podem ser utilizados instalados em uma luva. Estes atuadores são compostos por bolsas de ar que se inflam quando há contato com objetos do ambiente virtual, criando uma pressão sob os dedos do usuário (Figura 8.2). O mesmo sistema de colocação de atuadores

em uma luva pode ser utilizado para disponibilizar sensação térmica ou vibratória. Os atuadores térmicos, entretanto, possuem um funcionamento diferente e utilizam semicondutores para emitir a variação de temperatura ao usuário (Figura 8.3).
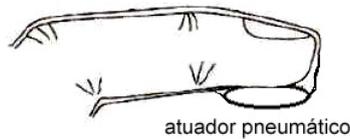
Figura 8.2. Exemplo de posicionamento de um atuador pneumático em uma luva para oferecer sensação tátil.
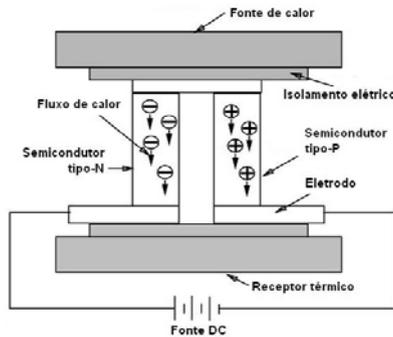
Figura 8.3. Esquema de um atuador termoelétrico. Adaptado de Burdea e Coiffet (2003), página 101.

Os atuadores vibratórios, por suas vez, utilizam pequenos motores elétricos, micro-auto-falantes ou mesmo materiais piezoelétricos [Khoudja et al., 2004] que emitem vibrações durante o contato com algum objeto. Tais atuadores são encontrados em luvas e em dispositivos 2D como o mouse (Figura 8.4a). A Figura 8.4b apresenta uma luva dotada de atuadores vibratórios localizados nas pontas dos dedos e na palma das mãos. A partir da detecção dos movimentos por sensores localizados nas juntas

espalhadas ao longo dos dedos e da mão, a luva é capaz de retornar as sensações de tato [Burdea e Coiffet, 2003].
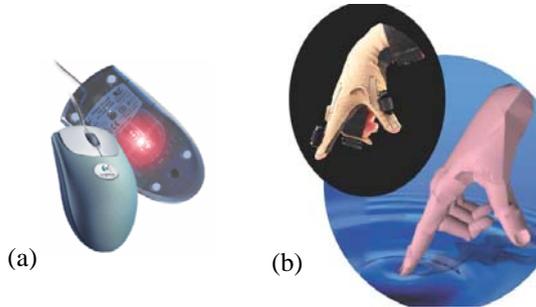
Figura 8.4. Dispositivos táteis baseados em atuadores vibratórios: (a) o mouse iFeel e (b) a luva CyberTouch. Fonte: http://www.logitech.com e http://www.immersion.com

Quando dotados de algum tipo de ponto de apoio, os dispositivos para as mãos podem oferecer *force-feedback*. O ponto de apoio é a estrutura responsável por oferecer resistência ao movimento do usuário. Atualmente, vários modelos de luvas são dotados desta capacidade. A diferença é o tipo de componente utilizado para oferecer resistência. A luva CyberGrasp, por exemplo, é a luva CyberTouch (Figura 8.4b) adicionada de um exoesqueleto [Immersion, 2007]. Um exoesqueleto também é usado pela luva Dextrous HandMaster (não oferece sensação tátil, apenas *force-feedback*). Uma outra luva, a Rutgers Master [Bouzit et al., 2002], baseia-se no uso de atuadores pneumáticos, compostos por pistões, que são os responsáveis pelas tensões fornecidas aos dedos do usuário através da compressão do ar. Uma das características deste dispositivo é a presença de uma única estrutura para medir a flexão dos dedos e oferecer as tensões. A Figura 8.5 apresenta a Rutgers Master e um exemplo de sua utilização em um ambiente de RV.
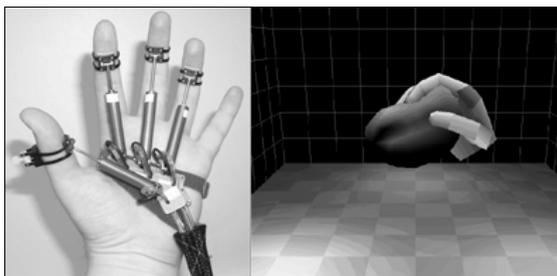
Figure 8.5. Luva com sistema unificado de captura de movimento e emissão de força com exemplo de aplicação. Fonte: Bouzit et al. (2002).

Mais recentemente, tem sido utilizada uma tecnologia baseada em pranchas para oferecer sensação tátil e *force-feedback*. Estas pranchas são dotadas de pinos ou jatos de ar que estimulam a superfície da pele do usuário. São utilizadas principalmente para a identificação de texturas ou transições entre objetos com a ponta dos dedos. Então, de acordo com a posição do dedo do usuário, pinos que se elevam ou ar jateado passam a fazer pressão no dedo [Khoudja et al., 2004]. A Figura 8.6 apresenta um dispositivo deste tipo.

Figure 8.6. Prancha de pinos para reconhecimento de texturas e transições entre objetos. Fonte: Khoudja et al. (2004).

8.2.2. Dispositivos para braços ou pernas

Os dispositivos para braços ou pernas visam oferecer, na sua grande maioria, *force-feedback*. Uma das razões deles não serem tão voltados para as sensações de tato é que as mãos são mais utilizadas e sensíveis para esta atividade. Desse modo, apesar de alguns oferecerem tato e *force-feedback*, sua principal função é restringir as ações do usuário, de acordo com suas ações, quando utilizados. Estes dispositivos costumam ter um custo proporcional à sua capacidade de suportar e retornar mais ou menos força. Apesar disso, há modelos desenvolvidos para uso popular, como mouse e *joystick*. Um exemplo é o mouse vibratório que, montado sobre uma base, é capaz de limitar as ações do usuário. Há alguns anos atrás a Logitech desenvolveu um mouse deste tipo chamado *Wingman* (Figura 8.7a) mas interrompeu sua produção em pouco tempo. Entretanto, diversos modelos de *joystick*, como o *SideWinder Pro*, são dotados desta capacidade e utilizados por fabricantes de jogos para aumentar o nível de envolvimento do usuário (Figura 8.7b). Neles são adicionando motores ou eixos à uma esfera localizada em sua base [Burdea e Coiffet, 2003].

Figura 8.7. Dispositivos de baixo custo com *force-feedback*. Fonte: fotos da autora.

Outra categoria de dispositivos disponíveis no mercado utiliza-se de uma espécie de braço mecânico para oferecer sensações de tato e *force-feedback*. Uma das vantagens deles é

permitir movimentos com 6DOF, bastante adequados a aplicações com visualização estereoscópica. O Phantom – *Personal Haptic Interface Mechanism*, da empresa Sensable, é um deles. Este dispositivo é composto por uma base à qual está atrelado o braço mecânico e cuja extremidade assemelha-se a uma caneta. Disponível em diferentes versões (Figura 8.8), utiliza potenciomêtros para oferecer *force-feedback* com diferentes intensidades em 3 (translações) ou 6 graus (translações + rotações) [Massie e Salisbury, 1994]. Sua popularidade deve-se à sua empunhadura, que permite naturalmente simular diversos instrumentos puntiformes. Princípio similar de funcionamento é usado na CyberForce, que acoplado à luva CyberTouch + CyberGrasp permite reconhecer objetos com tato e *force-feedback* para as mãos e para o braço [Immersion, 2006].

Figura 8.8. Duas versões do dispositivo PHANToM: Omni e Desktop 1.0. Fonte: fotos da autora.

Cabos metálicos são utilizados em outros dispositivos para também oferecer *force-feeback*. Nesta categoria, um pequeno objeto é conectado a cabos no interior de uma molduta. Estes cabos são atrelados a polias e são tensionados conforme os movimentos do usuário. A posição do objeto é identificada a partir do comprimento de cada cabo. Um dispositivo desta categoria é o SPIDAR – Space Interface Device for Virtual Reality [Sato, 2002]. Inicialmente desenvolvido para acoplar um único dedo, possui várias versões, inclusive para uso nas duas mãos. O esquema da Figura 8.9 apresenta a primeira versão do dispositivo, onde podem

ser vistos os atuadores e cabos utilizados para capturar os movimentos do usuário e oferecer *force-feedback*.
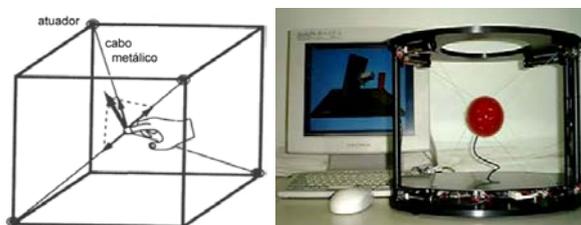
Figura 8.9. O dispositivo SPIDAR e sua versão com 7DOF. Fonte: Sato (2002).

Outro de dispositivo para braço é o Delta comercializado pela empresa Force Dimension (Figura 8.10). Fruto de um projeto de pesquisa francês, oferece nas suas versões mais recentes *force-feedback* com 6DOF de até 25N [Grange et al., 2001].

Figura 8.10. Dispositivo Delta. Fonte: Grange et al. (2001).

Além dos dispositivos citados, existem vários outros disponíveis no mercado. As principais diferenças entre eles, no entato, não são apenas quanto à tecnologia utilizada, mas também estão relacionadas ao espaço de atuação, à quantidade de força que são capazes de apresentar, à empunhadura e ao custo.

8.2.3. Dispositivos para o Corpo

Os dispositivos hápticos para o corpo são na realidade plataformas que visam oferecer vibrações ou mexer com a sensação de equilíbrio de seu usuário. Em alguns casos não são tratados como dispositivos hápticos pelo fato de não permitirem perceber objetos e sim mexerem com o equilíbrio do corpo. São comumente conhecidos como plataformas móveis e são largamente utilizados pela indústria do entretenimento na confecção de videogames. O VFlight é um exemplo destas plataformas e fornece movimentação em 185 graus [VFlight, 2007]. Outra plataforma utilizada para jogos é a X-Vector que oferece 360 graus de rotação para os eixos X e Z [EnTCo's, 2007]. A Figura 8.11 mostra uma imagem destas duas plataformas dedicada a jogos.

Figura 8.11. Plataformas móveis para jogos. Fonte: VFlight (2007) e EnTCo's (2007).

A aviação civil, por sua vez, faz uso destas plataformas há muitos anos para construir simuladores de vôo. Neste caso, a cabine é a do próprio avião simulado montada sobre uma plataforma móvel, composta por sistemas hidráulicos, acionada pelo sistema de RV que gera a simulação [Netto et al., 2002].

8.3 Pacotes de Programação

O controle de dispositivos hápticos depende do uso de pacotes de programação especificamente desenvolvidos para permitir gerar as

cenas hápticas. Este processo de geração é semelhante ao de desenvolvimento de uma cena gráfica, onde o programador deve definir o mundo (2D ou 3D), seus objetos e as propriedades destes. A diferença é que estas propriedades não irão se referir às características visuais do objeto, como a cor, mas sim às suas propriedades de material: rugosidade, elasticidade, etc. Assim, os pacotes de programação são constituídos por um conjunto de rotinas específicas para o controle dos dispostivos e envolvem, de modo geral, o reconhecimento do contato entre dispositivo (sua representação) e objetos, das características deste contato, o cálculo das forças de reação e o envio destas ao dispositivo.

Cada dispositivo disponível comercialmente costuma ter seu próprio pacote de programação que implementa as rotinas de processamento háptico. Um dos problemas relacionados à esta especificidade é que o programador de uma aplicação que envolva o uso de dispositivos hápticos deve aprender o pacote do dispositivo para poder utilizá-lo, o que acaba sendo uma árdua tarefa quando diferentes dispositivos são utilizados na aplicação.

Com o intuito de facilitar o processo de programação, tem sido desenvolvidos diferentes conjuntos de bibliotecas que abstraem as funcionalidades específicas de cada pacote de programação, de modo que as funções sejam únicas, independentes do dispositivo utilizado. O CyberMed é um destes pacotes [Souza et al., 2007]. Com distribuição livre e voltado para o sistema operacional Linux, o CyberMed contém um conjunto de classes C++ que permitem integrar sistemas hápticos em aplicações de RV. O Chai3D faz tarefa semelhante e, também com distribuição livre, é voltado ao sistema operacional Windows [Conti et al., 2003]. Um outro pacote desenvolvido e comercializado pela empresa ReachIn permite programadores C++, Phyton ou VRML desenvolverem aplicações sob Windows em rotinas pré-definidas [ReachIn, 2007].

8.4. Aplicações

A necessidade de oferecer sistemas mais realistas, fez com que uma série de aplicações passassem a ser desenvolvidas utilizando sistemas hápticos. Uma das áreas que vem utilizando com freqüência tais dispositivos é a medicina. A principal vantagem neste caso, é que assim torna-se possível simular procedimentos médicos com a finalidade de treinar novos profissionais. Nestes treinamentos, dispositivos hápticos simulam as sensações sentidas durante a manipulação das ferramentas médicas e auxiliam na aquisição de destreza manual, por exemplo. A Figura 8.12 apresenta uma aplicação desenvolvida pata o treinamento de cirurgias minimamente invasivas. Neste tipo de aplicação, dois dispositivos hápticos são utilizados na realização de um procedimento laparoscópico [Basdogan et al., 2004].

Figura 8.12. Simulação de laparoscopia. Fonte: Basdogan et al., (2007) e Burdea e Coiffet (2003) pg. 300.

As engenharias também tem se utilizado dos sistemas hápticos para auxiliar em tarefas de desenvolvimento e de manutenção. Na aplicação apresentada por Borro (2004), uma plataforma computacional oferece uma simulação de manutenção de aeronaves a partir de um modelo CAD. A Figura 8.13 mostra o sistema em uso e o esquema da plataforma utilizada.
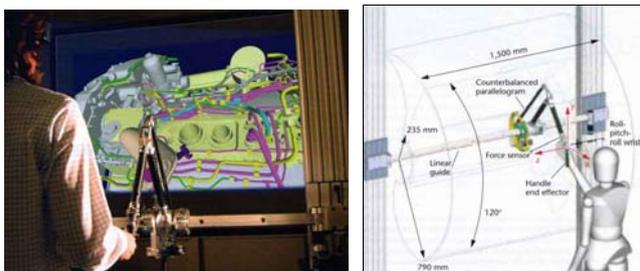
Figura 8.13. Manutenção de uma aeronave com auxílio de RV e sistemas hápticos. Fonte: Borro (2004).

Na aviação o uso de simuladores para o treinamento de pilotos têm sido largamente utilizado há vários anos. Atualmente, um piloto precisa ter um número mínimo de horas nestes simuladores para poder receber seu brevê. A empresa de aviação brasileira VARIG possui um destes sistemas para o modelo de avião Boeing 747. Neste simulador é posível simular pousos e decolagens com riqueza de sensações físicas (Figura 8.14).

Figura 8.14. Simulador de vôo da VARIG. Fonte: http://pauloribeiro.tripod.com/simulador.html. Acesso em março de 2007.

8.5. Referências Bibliográficas

Basdogan, C. et al. (2004) “Haptics in Minimally Invasive Surgical Simulation and Training”. IEEE CG&A, v. 24, n. 2, pp. 56-64.

Borro, D. et al. (2004) “A Large Haptic Device for Aircraft Engine Maintainability”. IEEE CG&A, v. 24, n. 6, pp. 70-74.

Bouzit, M.; Popescu, G.; Burdea, G. and Boian, R. (2002) “The Rutgers Master II-ND Force Feedback Glove”. Proc. 10th Symp. On Haptic Interfaces For Virtual Envir. & Teleoperator Systs. IEEE.

Burdea, G.; Coiffet, P. Virtual Reality Technology, 2ª edição. Wiley-Interscience, 2003.

Conti, F. et al. (2003) The CHAI libraries. *Eurohaptics*, 2003. Online: http://www.eurohaptics.vision.ee.ethz.ch/2003/75.pdf.

EnTCo's Monster Roll Cage Homepage. Online: http://mrc.idryonis.com/overview.php. Acesso em março de 2007.

Grange, S. et al. (2001) “Overview of the Delta Haptic Device”. Eurohaptics, 2001.

Immersion Corp. (2007) Online: http://www.immersion.com. Acesso em março de 2007.

Khoudja, M. et al. (2004) “Tactile interfaces: a state-of-the-art survey”. In: Proc. 35th Int. Symp. on Robotics. França.

Massie, T.; Salisbury, K. (1994)“PHANToM Haptic Interface: A Device for Probing Virtual Objects”. In: ASME Winter Annual Meeting, Vol 55-1, pp 295-300.

Netto, A.; Machado, L.; Oliveira, M. Realidade Virtual: Conceitos e Aplicações. Visual Books, 2002.

ReachIn Technologies (2007) “ReachIn API”. Online: http://www.reachin.se/products/ReachinAPI/. Acesso em março de 2007.

Sato, M. (2002) “SPIDAR and virtual reality”. In: Proceedings of the Fifth Biannual World Automation Congress, v. 13, pp. 17-23.

Seow, K. (1988) “Physiology of Touch, Grip and Gait”, In: J. Webster Ed., Tactile Sensing for Robotics and Medicine, John Wiley & Sons, New York, pp. 13-40.

Sherman, W e Craig, A., Understanding Virtual Reality. Morgan Kaufmann, 2003.

Souza, D.; Souza, L.; Cunha, I.; Moraes, R.; Machado, L. (2007) “Development of a VR Simulator for Medical Training Using Free Tools: A Case Study”. SVR'2007. (no prelo).

Tan, H. e Pentland, A. (2001) “Tactile Displays for Sensory Substitution and Wearable Computers”. In: Fundamentals of Wearable Computers and Augmented Reality, capítulo 18. Lawrence Erlbaum Associates, 2001.

VFlight (2007) Online: http://www.noonco.com/flyer/index.html. Acesso em março de 2007.

Capítulo 9

Ambientes Colaborativos de Realidade Virtual e Aumentada

Denise Filippo, Alberto Raposo, Markus Endler e Hugo Fuks

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Av. Marquês de São Vicente, 225 RDC – Gávea - 22453-900
Rio de Janeiro – RJ – Brazil

denise@les.inf.puc-rio.br, abraposo@tecgraf.puc-rio.br,
{endler, hugo}@inf.puc-rio.br

Abstract

This chapter deals with Virtual and Augmented Reality (VR and AR) Collaborative Environments from the point of view of collaborative systems, presenting them as a class of applications with specific workgroup-directed characteristics and demands. Environment exploiting the particularities of VR and AR applied to different collaborative scenarios are presented. Regarding trends, mobile supporting collaborative environments are emphasized.

Resumo

Este capítulo trata os Ambientes Colaborativos de Realidade Virtual (RV) e Realidade Aumentada (RA) sob a perspectiva de Sistemas Colaborativos, apresentando-os como uma classe de

aplicações com características, demandas e desafios específicos dos softwares direcionados para apoiar o trabalho em grupo. São apresentados e discutidos ambientes que exploram as particularidades das tecnologias de RV e RA quando aplicados a diferentes cenários de colaboração. No campo das tendências, é dada ênfase em ambientes colaborativos com suporte à mobilidade.

9.1. Introdução

A disseminação do uso dos computadores e da Internet fez surgir a questão de como estes recursos podem ser aproveitados para promover a colaboração entre usuários de grupos co-localizados e distribuídos. Com este objetivo, são desenvolvidas pesquisas na área de Trabalho Colaborativo Apoiado por Computador (CSCW - *Computer Supported Collaborative Work*), uma área de estudo interessada no trabalho em conjunto de grupos de pessoas com o apoio de computadores. Groupware (sistema ou ambiente colaborativo) é a tecnologia baseada em mídia digital que dá suporte às atividades de pessoas organizadas em grupos. Aplicações de bate-papo, mensagem instantânea, sites de relacionamentos, blogs, sistemas de recomendação e ferramentas de compartilhamento de arquivos são exemplos de groupware.

Neste contexto, os Ambientes Colaborativos de RV e RA abrem uma nova perspectiva para a colaboração em grupo, ao possibilitar que seus participantes interajam através da simulação de um mundo real ou imaginário ou através da manipulação de objetos virtuais no mundo real. O objetivo deste capítulo é, por um lado, apresentar o que as tecnologias de RV e RA podem agregar aos sistemas colaborativos de hoje em dia e, por outro lado, discutir como as aplicações de RV e RA podem ser beneficiadas pelos avanços na área de sistemas colaborativos.

Na seção 9.2 é apresentada uma visão geral da Colaboração e dos sistemas colaborativos. As seções 9.3 e 9.4 abordam, respectivamente, as características dos ambientes colaborativos de RV e de RA. Na seção 9.5 são apresentadas tendências para estes ambientes. A seção 9.6 conclui este capítulo.

9.2. Sistemas Colaborativos

Neste capítulo são introduzidos os conceitos básicos de Colaboração e, em especial, o relacionamento dos sistemas colaborativos com a RV e RA. O Modelo 3C de Colaboração e possíveis classificações para estes sistemas são apresentados.

9.2.1. Dimensões da colaboração

Colaboração designa a ação de trabalhar em conjunto ou a realização de um trabalho em comum com uma ou mais pessoas [Ferreira, 1986]. Para que haja colaboração, os membros de uma equipe precisam trocar informações (se comunicar), organizar-se (se coordenar) e operar em conjunto num espaço compartilhado (cooperar). Baseado nestes conceitos, na Figura 9.1 é apresentado o diagrama do Modelo 3C de Colaboração [Fuks et al, 2005], um refinamento do modelo proposto por Ellis, Gibbs e Rein (1991).

Figura 9.1 - Modelo 3C de Colaboração.

Comunicação é a ação de trocar mensagens para que haja entendimento comum das idéias discutidas. Na Colaboração, a Comunicação é direcionada para a ação [Fuks et al., 2003]. Os participantes de um grupo de trabalho trocam idéias e pontos de vista, apresentam seus argumentos e discutem com a finalidade de realizar negociações, tomar decisões e firmar compromissos.

Coordenação é a ação de dispor segundo uma certa ordem e método, organizar, arranjar. A Coordenação também está associada à idéia de planejamento. Fazem parte da Coordenação a identificação de objetivos, o mapeamento dos objetivos em tarefas, a seleção dos participantes e a distribuição das tarefas. Através da Coordenação, as tarefas atribuídas aos diferentes participantes são realizadas na ordem e no tempo previstos e de acordo com os objetivos e restrições determinados [Raposo e Fuks, 2002].

Cooperação é a ação de operar simultaneamente. Ao cooperarem, os participantes do grupo produzem, modificam e utilizam de maneira compartilhada um conjunto informações e artefatos reais ou virtuais, tais como um documento, uma planilha, ou a peça de um carro. A Cooperação é então esta operação conjunta dos membros do grupo no espaço compartilhado a fim de que as atividades possam ser realizadas [Fuks et al., 2002].

O Modelo 3C evidencia a necessidade de os participantes terem informações sobre o que está ocorrendo. Através das informações de percepção disponibilizadas pelo ambiente, os participantes de um grupo tomam ciência, por exemplo, de quais etapas já foram terminadas e de quem é a responsabilidade de uma tarefa. Informações como estas são necessárias para que cada participante seja capaz de avaliar seu trabalho e dos demais participantes e de redirecionar suas atividades se necessário.

9.2.2. Sistemas colaborativos e sua classificação

DeSanctis e Gallupe (1987) utilizam a noção de tempo e de espaço, entendidos como 2 eixos, para classificar um sistema colaborativo. O eixo do tempo indica se os participantes interagem de maneira síncrona ou assíncrona, enquanto o eixo do espaço indica se eles estão fisicamente próximos ou não. Quando os interlocutores precisam estar simultaneamente conectados ao ambiente para que a colaboração se efetive, o sistema é considerado síncrono. Quando esta simultaneidade não for necessária, o sistema é considerado assíncrono. Na figura 9.2 são mostrados exemplos de sistemas colaborativos classificados de acordo com seu posicionamento nos 2 eixos de tempo e espaço.

ESPAÇO \ TEMPO	mesmo tempo (síncrono)	tempo diferente (assíncrono)
mesmo lugar (local)	**Interações síncronas face-a-face** *Brainstorming*	**Interações assíncronas locais** *Post-It Notes*
local diferente (distribuído)	**Interações síncronas distribuídas** Bate-papo Videoconferência	**Interações assíncronas distribuídas** Correio eletrônico Fórum

Figura 9.2 – Exemplo da classificação dos sistemas colaborativos em função de tempo e espaço.

Os diferentes sistemas colaborativos também são classificados quanto à sua maior tendência de oferecer itens de Comunicação, Coordenação e Cooperação (Teufel et al., 1995; Borghoff e Schlichter, 2000). A Figura 9.3 esquematiza como esta classificação pode ser feita.

Figura 9.3 - Classificação dos sistemas colaborativos de acordo com as dimensões do Modelo 3C.

Deve-se notar que o fato de uma aplicação ser classificada segundo uma das 3 dimensões do Modelo 3C não implica que ele não contenha funcionalidades das demais dimensões.

9.2.3. Realidade virtual e aumentada para a colaboração

A idéia de colaborar em espaços virtuais não é nova. Sistemas experimentais utilizando RV para Colaboração são usados há décadas, mas apenas recentemente começaram a sair das esferas acadêmicas e militares. Esse aumento de popularidade se deve principalmente ao aumento da capacidade de processamento das máquinas e sua redução de custos, bem como a disponibilidade cada vez maior de redes com alta velocidade. Assim, ambientes virtuais colaborativos têm conseguido bastante popularidade com os jogos multi-usuários e comunidades virtuais, tais como o Second Life [Secondlife, 2007].

Em paralelo, o desenvolvimento de sistemas de RA também se volta para aplicações colaborativas. Com a RA, usuários co-localizados podem compartilhar objetos reais e virtuais em meio a um espaço comum no mundo real, não precisando estar diante de um computador. A introdução de objetos virtuais no espaço compartilhado é útil não só para possibilitar que o grupo manipule objetos inexistentes no mundo real, mas também para aumentar os objetos reais com informações que são consultadas e modificadas pelos usuários à medida que o trabalho progride.

Apesar da crescente popularidade, sistemas colaborativos de RV e RA ainda apresentam uma série de desafios em seu desenvolvimento. Entre esses desafios, podem ser mencionados os problemas relacionados ao gerenciamento de recursos de rede, ao processamento gráfico em tempo real e à sincronização de visão em aplicações multi-usuários (por exemplo, manutenção de consistência do estado de objetos compartilhados entre os usuários). Existem também as dificuldades específicas de RV e RA. Em RV, por exemplo, há a dificuldade de se manipular os objetos do mundo virtual e a necessidade de se criar avatares realistas para ampliar tanto a capacidade de comunicação entre os participantes quanto o sentimento de presença no grupo. Em RA, por exemplo, há o problema da precisão de registro (i.e., alinhamento preciso dos objetos virtuais sobre os objetos reais) e do desenvolvimento de dispositivos de apresentação para misturar os objetos virtuais e reais [Filippo et al., 2005]. Somado a tudo isso, há também as dificuldades específicas da área de aplicação, tais como a integração com grandes bases de dados (por exemplo, para as informações geográficas de simulações militares) e questões de segurança (para aplicações de comércio eletrônico, por exemplo).

Normalmente, a literatura de ambientes virtuais colaborativos enfoca os problemas ligados à infra-estrutura

tecnológica para o desenvolvimento dos mesmos [Singhal e Zyda, 1999; Macedônia e Zyda, 1997; Rodrigues et al., 2004]. Neste capítulo, os ambientes colaborativos de RV e RA são analisados conforme suas características para Colaboração apresentadas nas seções anteriores.

9.3. Ambientes Colaborativos de Realidade Virtual

Os ambientes virtuais colaborativos (CVEs – *Collaborative Virtual Environments*) apresentam grande potencial para o suporte ao trabalho colaborativo. Um CVE é definido como uma simulação em tempo real de um mundo real ou imaginário, onde usuários estão simultaneamente presentes e podem navegar e interagir com objetos e outros usuários [Hagsand, 1996]. Em CVEs, os usuários são representados por avatares (Figura 9.4).

De uma maneira geral, os CVEs são usados para Colaboração com usuários localizados remotamente e, do ponto de vista temporal, podem ser tanto síncronos quanto assíncronos. Nos últimos anos, têm sido realizados trabalhos no sentido de desenvolver CVEs para usuários no mesmo local, criando ambientes "fisicamente compartilhados" [Mulder e Boschker, 2004; Kitamura et al., 2001]. Porém, este capítulo se restringirá aos CVEs para usuários remotamente localizados.

Figura 9.4 – Exemplo de CVE: Walkinside Viewer, para visualização de modelos CAD de engenharia [VRContext, 2007].

CVEs também apresentam características bastante interessantes no que diz respeito a Comunicação, Coordenação e Cooperação. A Comunicação geralmente é realizada através de ferramentas de bate-papo, que acompanham os CVEs mais simples, ou na forma de áudio e videoconferência, que acompanham os CVEs mais sofisticados. A Coordenação é elemento-chave em CVEs usados para fins específicos, como os militares e os jogos multi-usuários. A Cooperação, por sua vez, faz parte da própria natureza dos CVEs, que são literalmente espaços de trabalho compartilhados.

Nas próximas seções serão apresentados alguns CVEs, classificados de acordo com seu propósito principal, dentre as dimensões do Modelo 3C de Colaboração.

9.3.1. Comunicação em CVEs

Em CVEs, os usuários, na forma de seus avatares, são livres para navegar no espaço virtual, encontrando-se uns com os outros.

Dessa forma, um canal de comunicação verbal ou não-verbal é uma necessidade primária em CVEs. A comunicação verbal está presente em chats e canais de áudio e vídeo que podem acompanhar o CVE. Além dessas possibilidades, que também são típicas de aplicações colaborativas de desktop, a Comunicação é ampliada em CVEs pela capacidade de comunicação não-verbal, tais como gestos, expressões faciais, e postura do avatar.

A noção de espaço e a metáfora de mundo real oferecidas por CVEs também trazem possibilidades de percepção que não são triviais em aplicações de desktop. Por exemplo, a movimentação dos avatares antecipa intenções dos usuários e revela a expectativa que os demais usuários têm a seu respeito (por exemplo, um avatar se aproximando do outro pode indicar que ele deseja se comunicar).

Os CVEs voltados para Comunicação são plataformas de interação social, onde os usuários normalmente se encontram casualmente e se comunicam. Um exemplo de CVE focado na Comunicação é o Active Worlds [Active Worlds, 2007], uma ferramenta para a criação de chats 3D na Web.

Outro CVE voltado para a Comunicação é o FreeWalk [Nakanishi, 2004]. Em sua primeira versão, o FreeWalk apresentava-se como uma videoconferência espacial, voltada para o suporte a encontro casuais. Cada avatar é representado por uma janela 2D no ambiente 3D, que transmite seu áudio e vídeo aos demais. A noção espacial acrescenta os conceitos de localização e direção do olhar às videoconferências tradicionais. Em sua segunda versão, foi acrescentado um agente autônomo participante do ambiente, cujo objetivo é facilitar a encontro casual entre estranhos (recurso de Coordenação). Em sua terceira versão, o sistema apresenta simulações de situações reais: uma situação de desastre

onde os participantes (humanos e agentes) assumem papéis em um processo de evacuação (foco na Cooperação).

Apesar de poder ser feita de forma assíncrona (devido à possibilidade de alterar e deixar recados em artefatos do mundo virtual), a Comunicação nos CVEs geralmente é síncrona.

9.3.2. Coordenação em CVEs

Além da potencialidade dos CVEs para a Comunicação, também é reconhecido seu potencial para a Colaboração baseada em artefatos (Cooperação), uma vez que CVEs disponibilizam objetos e dados no espaço virtual [Zhang e Furnas, 2002]. Coordenação, dentre as 3 dimensões da colaboração, é ainda a menos explorada nos CVEs.

Entretanto, CVEs voltados para simulações militares e jogos multi-usuários são exemplos de aplicações colaborativas que necessitam de sofisticados mecanismos de Coordenação para garantir a realização das tarefas [Capps et al., 2001].

Um uso relativamente comum de CVEs que permite classificá-los como ferramenta focada em Coordenação é o treinamento. O SAVE [Holm et al., 2002], por exemplo, é uma aplicação para treinar trabalhadores em situações de emergência em refinarias de petróleo. Tais situações são simuladas no ambiente virtual e o treinamento usa uma metáfora de "Deus Grego", onde a pessoa a ser treinada é o "herói", que realizará as tarefas no ambiente virtual, e o instrutor é o "Deus no Olimpo", que coordena toda a simulação (altera parâmetros, move objetos e o herói, etc)

9.3.3. Cooperação em CVEs

Cooperação é a dimensão da colaboração mais próxima dos CVEs. Mesmo nos CVEs focados na Comunicação e na Coordenação, a Cooperação está sempre materializada no espaço virtual, onde

ocorrem as interações entre os usuários. Por outro lado, é possível conceber CVEs apenas com serviços de Cooperação, como é o caso do Walkinside Viewer [VRContext, 2007], ilustrado na Figura 4. Neste CVE, vários usuários podem acessar o ambiente virtual (na verdade, um modelo CAD) e navegar por ele, sem qualquer recurso de Comunicação ou Coordenação (assume-se que isto é feito por meios externos à aplicação, como o telefone).

CVEs trazem novas possibilidades a vários aspectos essenciais à Cooperação, tais como a noção de contexto compartilhado e a percepção mútua entre os usuários [Snowdon et al., 2001]. A noção de contexto compartilhado está relacionada às informações perceptuais dos objetos e eventos que podem ser explorados e manipulados no espaço de trabalho. Por exemplo, quando a Cooperação se baseia em artefatos (documentos compartilhados, por exemplo), é a visualização das alterações nesses artefatos que direciona as ações futuras dos usuários. A percepção está ligada a "entender as atividades dos outros para prover o contexto para suas próprias atividades" [Dourish e Bellotti, 1992]. Nesses aspectos, a representação dos avatares e os recursos de visualização no espaço 3D podem ampliar a capacidade de entender o que os outros usuários estão fazendo, em que objetos eles estão atuando, o que se espera que eles vão fazer em seguida (por exemplo, se eles estão indo para outro lugar), etc. Esta capacidade dos CVEs está ligada ao fato deles estarem mais próximos da representação do mundo real do que as aplicações de desktop, onde a transmissão de todas essas informações não é tão natural.

9.4. Ambientes Colaborativos de Realidade Aumentada

Azuma (2001) define um sistema de Realidade Aumentada como sendo aquele que possui as seguintes características: combina objetos reais e virtuais num ambiente real; opera interativamente e

em tempo real, e registra (alinha) objetos reais e virtuais uns com os outros. Uma vez que a RA é, por definição, fortemente direcionada para a apresentação e manipulação de objetos virtuais, os ambientes prestam-se naturalmente ao compartilhamento destes objetos e, consequentemente, ao suporte à Cooperação (Figura 9.5). Quando estes objetos são utilizados para orientar e direcionar o usuário nas atividades do seu trabalho, os sistemas de RA estão provendo suporte à Coordenação. A Comunicação através de objetos virtuais é a menos explorada em ambientes colaborativos de RA, pois estes aproveitam as tecnologias já mais estabelecidas de áudio e vídeo-conferência.

Figura 9.5 – Exemplo de ambiente colaborativo de RA.

Para Billinghurst and Kato (2002), a principal característica de ambiente colaborativos de RA é a natureza “sem cortes” da interface RA, onde usuários vêem-se uns aos outros ao mesmo tempo em que vêem objetos virtuais em meio a eles. A transição entre o real e o virtual é mais suave, não ficando delimitada pela tela do computador ou pelo mundo virtual dos CVE. Billinghurst and Kato também ressaltam que as interfaces RA para usuários co-localizados, ao contrário de outras tecnologias de CSCW, não fazem a separação entre o espaço de comunicação e

o espaço de tarefas (sob a ótica do Modelo 3C, espaço de Cooperação).

Em RA, diferentes objetos virtuais podem ser apresentados para diferentes usuários, possibilitando a existência de espaços públicos e privados e de diferentes visões do espaço compartilhado. Por exemplo, enquanto um usuário vê sobre uma parede a tubulação de água, o outro vê o cabeamento elétrico.

As diferentes visões do espaço compartilhado em ambientes colaborativos de RA co-localizados introduz o problema de se ter que garantir um entendimento comum sobre este espaço [Azuma, 2001]. Deve-se evitar, por exemplo, que um usuário aponte para um objeto e o outro usuário veja um objeto diferente naquele lugar. Outras questões a serem tratadas são compartilhamento e colisão de objetos virtuais e o fato de que o uso de capacetes ou óculos do tipo vídeo-through em ambientes co-localizados ou em vídeo-conferências impede que os usuários vejam os olhos uns dos outros.

9.4.1. Comunicação em RA

A operação em tempo real característica da RA, presente em sua própria definição, reflete-se no uso freqüente de comunicação síncrona nestes ambientes. Nos casos em que os usuários estão co-localizados, a comunicação é realizada diretamente, sem necessidade do ambiente para intermediá-la. No caso de se ter interlocutores fisicamente distantes, audio-conferência [NetAttack, 2007] e vídeo-conferência [Etälä, 2007] têm sido a solução adotada. Observa-se, porém, que no áudio e na videoconferência a troca de mensagens não faz uso de objetos virtuais, isto é, nestes casos a Comunicação não é baseada na tecnologia de RA.

Um exemplo clássico sistema de comunicação de RA propriamente dito é apresentado no filme Guerra nas Estrelas de George Lucas. Neste filme, os personagens conversam com interlocutores remotos apresentados como uma imagem virtual de corpo inteiro e em tamanho real. Um sistema de videoconferência RA que segue esta abordagem é apresentado em [Billinghurst, et al., 2002]. Neste sistema, o rosto do interlocutor remoto é apresentado como um objeto virtual sobre um marcador. Vários interlocutores, móveis ou em desktops, podem participar da conferência simultaneamente. Os marcadores funcionam como monitores virtuais, com a vantagem de poderem ser rearrumados livremente na mesa de trabalho e de serem carregados por usuários móveis. Um ambiente de Comunicação RA como este evita a limitações da áudio conferência, como falta de visão dos interlocutores, da vídeo-conferência, como dificuldade de capturar movimentos sutis e de obter informações sobre o espaço em que estão os participantes, e dos CVE, como desconexão do mundo real, mas ainda requer melhorias devido à necessidade de uso do capacete e de rastreamento e registro [Billinghurste e Kato, 2002].

Outra opção para a realização de comunicação síncrona é o uso de quadros-brancos virtuais [Etälä, 2007]. Embora este seja um artefato destinado para desenhar esboços e esquemas, ele também é usado como suporte para a troca de mensagens.

Os ambientes de RA já apresentam diferentes soluções para a comunicação em ambientes síncronos. Estas podem ser estendidas para os cenários assíncronos: nos exemplos acima, disponibilizando um replay da imagem dos interlocutores falando uma mensagem ou mantendo entre sessões de trabalho as mensagens escritas no quadro-branco.

9.4.2. Coordenação em RA

No contexto dos ambientes colaborativos de RA, serviços de Coordenação são usualmente oferecidos em conjunto com outros serviços e diferenciam-se das aplicações típicas de Coordenação oferecidas nos desktops, como workflow e agenda compartilhada.

Este é o caso dos ambientes MARS - Mobile Augmented Reality Systems [Feiner, 2006] e OCAR - Outdoor Collaborative Augmented Reality [Reitmayr, 2004], que são sistemas colaborativos móveis, sensíveis à localização, em que usuários visitam uma área externa e consultam informações sobre ela. Nestes ambientes, objetos virtuais como bandeiras, caminhos e estacas são utilizados para que usuários conhecedores do ambiente coordenem os visitantes até que estes cheguem ao seu destino. Outro exemplo de suporte à Coordenação está presente em funcionalidades do sistema de busca de informações do OCAR. Esta aplicação possibilita que um guia de turismo coordene um grupo de turistas, selecionando e disparando em seus visores objetos virtuais que chamam a atenção para pontos de interesse que devem ser observados a cada momento.

Assim como no caso dos sistemas focados na Comunicação, ambientes colaborativos de RA co-localizados ou em ambientes remotos não precisam necessariamente prover mecanismos de Coordenação explícitos. A Coordenação de um grupo pode se dar através do protocolo social, isto é, através da habilidade dos participantes de mediar as interações.

9.4.3. Cooperação em RA

Assim como nos CVEs, os ambientes colaborativos de RA estão fortemente associados à Cooperação. Na essência da tecnologia RA está a apresentação e manipulação de objetos virtuais no mundo real, provendo o suporte necessário para que a operação conjunta

no espaço compartilhado (Cooperação) ocorra. Mais que isto, este espaço compartilhado é o mundo real e a manipulação dos objetos não é necessariamente delimitada por uma tela de monitor nem precisa ficar presa em torno de uma mesa, abrindo novas possibilidades de suporte à Cooperação.

A maioria dos ambientes colaborativos de RA trabalha com o aumento do sentido da visão em cenários síncronos co-localizados ou remotos. Uma exceção é o AGroove, voltado para a criação conjunta de música com instrumentos reais e virtuais [Poupyrev et al. 2000]. Há ainda algumas aplicações que fazem uso simultâneo de RA e RV [Magic Book, 2007; Etälä, 2007].

A visualização dos diferentes participantes de um grupo pode se dar de diferentes formas. Em ambientes co-localizados, usuários com capacetes ou óculos especiais podem trabalhar em grupo com objetos virtuais como se eles existissem num mesmo ponto do espaço físico [Billinghurst et al., 2001, Kaufmann, 2006]. Outra maneira é cada participante ver os objetos de trabalho através do visor do PDA, celular ou monitor [Wagner et al., 2005; Wagner et al., 2006]. A tecnologia de projeção possibilita que o grupo opere com objetos físicos sobre os quais são projetados objetos virtuais, além de prover meios para que objetos reais sejam camuflados [Inami et al., 2000]. No caso de usuários remotos, não há alternativa: as imagens do mundo real têm que ser transmitidas e, junto com as imagens dos objetos virtuais, devem ser visualizadas através de uma tela de vídeo [Feiner, 2006; Etälä, 2007].

9.5. Conclusões e Tendências

Considerando o Modelo 3C de Colaboração, a Comunicação apresenta-se como elemento importante nos ambientes de RV, visto que eles são desenvolvidos em sua maioria para usuários

remotamente localizados, o que não acontece na maioria dos sistemas colaborativos de RA. Já os serviços de Coordenação, quando existentes, são usualmente oferecidos em conjunto com outros tipos de serviços. O grande potencial dos sistemas colaborativos de RV e RA, principalmente se comparado aos sistemas em desktop, está no suporte à Cooperação, tendo em vista que eles têm como base a interação e manipulação de objetos em espaços compartilhados.

Os ambientes colaborativos de RV e RA são, por si só, uma tendência nos sistemas de software e, da mesma forma, os sistemas colaborativos fazem parte cada vez mais do dia-a-dia das pessoas. A difusão e, principalmente, a fusão destas tecnologias abrirá novas oportunidades para que participantes de um grupo trabalhem em conjunto, pois possibilitam que este grupo interaja no ambiente real ou em simulações de ambientes tridimensionais, ao invés das telas bidimensionais do computador. Desta forma aumenta-se o grau de percepção mútua e de compartilhamento de contexto, fatores importantes para que a Colaboração se realize.

Mark Weiser, considerado o pai da Computação Ubíqua, observou que a Realidade Virtual apresenta-se como o oposto da Computação Ubíqua: segundo ele, a Realidade Virtual tenta colocar o mundo real dentro do computador, simulando-o parcialmente e envolvendo o usuário neste mundo virtualizado, enquanto na Computação Ubíqua são os equipamentos computacionais que permeiam o mundo real do usuário, enriquecendo-o [Weiser, 1991]. É interessante, portanto, situar a Realidade Aumentada sob esta ótica: ao utilizar a tecnologia da Realidade Virtual no mundo real, a Realidade Aumentada, especialmente no caso dos sistemas que fazem uso de equipamentos móveis, também enriquece e permeia o ambiente real. Uma vez que a tecnologia produz equipamentos cada vez menores, mais poderosos, mais portáteis, "vestíveis" e com

capacidade de estarem continuamente conectados, a tendência é que um grande número dos sistemas de Realidade Aumentada se tornem Sistemas de Realidade Aumentada Ubíqua. Desta forma, a Realidade Aumentada Ubíqua deverá diminuir cada vez mais a distância entre a Realidade Virtual e a Computação Ubíqua (mundo real), no sentido de que a transição entre os extremos do Contínuo de Milgram [Milgram, 1994] irá se tornar cada vez mais suave e transparente ou, como na proposta de Tecnologia Calma de Weiser [Weiser, 1995], mais calma e invisível.

Com a possibilidade de mobilidade dos usuários, outra tendência dos ambientes de RV e RA é o das aplicações sensíveis à localização. O uso de serviços e informações disponibilizados em função da localização do usuário apresenta grande potencial para novas formas de Colaboração. Em ambientes como o OCAR e MARS, o usuário tem acesso à informação no próprio local, ancorado ao mundo físico e com possibilidade de interagir e de modificá-la. Ao observar isto, Höllerer [1999] chega a afirmar que "o mundo se torna a interface". Estes sistemas podem significar o início do WorldBoard proposto por Spohorer [1997]: um hipertexto espacial, com abrangência mundial, de informações ancoradas a locais e objetos físicos.

9.6. Referências Bibliográficas

Active Worlds, http://www.activeworlds.com/ Acesso em Jan/2007

Azuma, R.; Baillot, Y.; Behringer, R.; Feiner, S.; Julier, S.; Macintyre, B. (2001) "Recent advances in augmented reality". IEEE Computer Graphics and Applications, v.21, n.6, Nov/Dez, pp 34-47

Billinghurst, M. Cheok, A. Prince, S. Kato, H. (2002) "Real world teleconferencing". Computer Graphics and Applications, IEEE, v.22, n.6, Nov/Dez, ISSN: 0272-1716, pp 11-13

Billinghurst, M.; Kato, H. (2002) "Collaborative Augmented Reality". Communications of the ACM, Vol. 45, No. 7, July, ISSN:0001-0782, pp.64 – 70

Billinghurst, M., Kato, H., Poupyrev, I. (2001) The MagicBook: Moving Seamlessly between Reality and Virtuality, IEEE Computer Graphics and Applications, Maio/Jun, pp. 2-4

Borghoff, U.M.; Schlichter, J.H. (2000) “Computer-Supported Cooperative Work: Introduction to Distributed Applications”, Springer, USA.

DeSanctis, G.; Gallupe, B. (1987) “A foundation for the study of group decision support systems”, Management Science, v. 33, n. 5, pp. 589-609

Dourish, P.; Belloti, V. (1992) “Awareness and Coordination in Shared Workspaces” Conference on Computer-Supported Cooperative Work (CSCW), eds: Turner e Kraut, pp. 107-114.

Ellis, C. A., Gibbs S. J.; Rein, G. L. (1991) “Groupware – Some Issues and Experiences”, Communications of the AM, Jan, vol 34, n.1, pp 38-58

Etälä - http://www.automation.hut.fi/projects/etala/index.html Acesso em: Jan/2007.

Feiner, S.; Höllerer T.; Gagas, E.; Hallaway, D.; Terauchi T.; Güven, S.; MacIntyre B. “MARS – Mobile Augmented Reality System” http://www1.cs.columbia.edu/graphics/projects/mars, Acesso em Jan/2007

Ferreira, A.B.H. (1986) Novo Dicionário da língua portuguesa. 2. ed. Rio de Janeiro, Nova Fronteira.

Filippo, D., Endler, M.; Fuks, H. “Colaboração móvel com realidade aumentada”. Monografias em Ciência da Computação 01/05 Ed: Carlos José Pereira de Lucena, Jan 2005, ISSN 0103-9741, 33 p., http://bib-di.inf.puc-rio.br/techreports/2005.htm

Fuks, H., Raposo, A.; Gerosa, M.A. (2002) "Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas", XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação, v.2, Cap. 3, ISBN 85-88442-24-8, pp 89-128.

Fuks, H., Gerosa, M.A.; Pimentel, M. (2003) "Projeto de Comunicação em Groupware: Desenvolvimento, Interface e Utilização", XXII Jornada de Atualização em Informática, Anais do XXIII Congresso da Sociedade Brasileira de Computação, v.2, cap. 7, ISBN 85-88442-59-0, pp. 295-338.

Fuks, H., Raposo, A., Gerosa, M.A.; Lucena, C.J.P. (2005) Applying the 3C Model to Groupware Development. International Journal of Cooperative Information Systems (IJCIS), v.14, n.2-3, Jun-Sep 2005, World Scientific, ISSN 0218-8430, pp 299-328

Hagsand, O. "Interactive Multiuser VEs in the DIVE System", IEEE Multimedia, 3(1):30-39. Spring 1996.

Höllerer T.; Feiner S.; Terauchi T.; Rashid, G.; Hallaway, D. (1999) "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system" Computers and Graphics, v. 23, n.6, Elsevier Publishers, Dez, p. 779-785

Holm, R., Priglinger, M. , Stauder, E., Volkert, J., Wagner. R. (2002) "A Combined Immersive and Desktop Authoring Tool for Virtual Environments". IEEE Virtual Reality 2002, USA, pp. 93-100.

Inami M., Kawakami, N., Sekiguchi, D. Yanagida, Y. (2000) "Visuo-Haptic Display Using Head-Mounted Projector" IEEE Virtual Reality 2000, USA, pp. 233-240.

Kaufmann, H. (2006) "The potential of augmented reality in dynamic geometry education", 12th International Conference On Geometry and Graphics (ISGG), Ago 6-10, Salvador, Brasil

Kitamura, Y., Konishi, T., Yamamoto, S.; Kishino, F. (2001) "Interactive stereoscopic display for three or more users", Computer Graphics (SIGGRAPH 2001 Proceedings), v.35, pp 231–239.

Macedonia, M. R.; Zyda, M. J. (1997) "A Taxonomy for Networked Virtual Environments. IEEE Multimedia", vol. 4 n.1 Jan/Mar, pp 48-56

Capps, M.; McDowell, P.; Zyda, M. (2001) "A future for entertainment-defense research collaboration," IEEE Computer Graphics and Applications, v. 21, n.1, pp. 37–43

Milgram, P.; Kishino, F. (1994) "A taxonomy of mixed reality visual displays", IEICE Transactions on Information Systems, v. E77-D, n.12, Dez, pp. 1321-1329

Mulder, J. D.; Boschker, B. R. (2004) "A Modular System for Collaborative Desktop VR/AR with a Shared Workspace", IEEE Conference on Virtual Reality (VR' 04), Amsterdam, pp. 75-82.

Nakanishi, H. (2004) "FreeWalk: a social interaction platform for group behaviour in a virtual space". International Journal of Human-Computer Studies, v.60, n.4, April, pp. 421-454.

NetAttack - http://www.fit.fraunhofer.de/projekte/netattack/index_en.xml?aspect=spielidee, Acesso em Jan/2007

Poupyrev, I., Berry, R., Kurumisawa, J., Nakao, K., Billinghurst, M., Airola, C., Kato, H., Yonezawa, T. e Baldwin, L. (2000) "Augmented Groove: Collaborative Jamming in Augmented Reality", SIGGRAPH' 2000 Conference Abstracts and Applications, pp. 77

Raposo, A.; Fuks, H. (2002) "Defining Task Interdependencies and Coordination Mechanisms for Collaborative Systems". in: Blay-Fornarino, M., Pinna-Dery, A. M., Schmidt, K. & Zaraté, P.; Cooperative Systems Design (vol 74 of Frontiers in Artificial

Intelligence and Applications), ISBN 1-58603-244-5 IOS Press, Amsterdam, pp 88-103.

Reitmayr, G.; Schmalstieg, D. (2004) “Collaborative augmented reality for outdoor navigation and information browsing”, 2nd Symposium on Location Based Services and TeleCartography, Viena, Jan, p. 53 - 62

Rodrigues, L. C. R., Kubo, M. M., Rodello, I. A., Sementille, A. C., Tori, R., Brega, J. R. F. (2004) “Ambientes Virtuais Distribuídos”. Realidade Virtual: Conceitos e Tendências – Pré-Simpósio SVR 2004, C. Kirner & R. Tori (eds.), SBC, pp 43-59

Second Life - http://secondlife.com/ Acesso em Jan/2007

Singhal, S.; Zyda, M. (1999) “Networked Virtual Environments: Design and Implementation”, Addison-Wesley, USA, 1999. ISBN 0-201-32557-8

Snowdon, D., Churchill, E. F.; Munro, A. J. (2001) "Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction", Collaborative Virtual Environments: Digital Places and Spaces for Interaction, E. F. Churchill, D. Snowdon and A. J. Munro (eds), cap 1, Springer-Verlag, pp. 3-17

Spohrer, J. (1997) What comes after the WWW?. ISITalk 1997 http://www.worldboard.org/pub/spohrer/wbconcept/default.html Acesso em Jan.2007

Teufel, S., Sauter, C., Mühlherr, T. e Bauknecht, K. (1995) “Computerunterstützte Gruppenarbeit”. Bonn: Addison-Wesley

Wagner, D. Pintaric, T. Ledermann, F.; Schmalstieg, D. (2005) “Towards Massively Multi-User Augmented Reality on Handheld Devices”. 3rd International Conference on Pervasive Computing (Pervasive 2005), Munique

Wagner, D., Schmalstieg, D., Billinghurst, M.. (2006) Handheld AR for Collaborative Edutainment. In Proceedings of the 16th

International Conference on Artificial Reality and Telexistence (ICAT 2006), Nov, 29 – Dec,1, Hangzhou, China.

Weiser, M. (1991) "The computer for the twenty-first century", Scientific American, Set, pp. 94-100

Weiser, M. (1995) Designing Calm Technology, Dez, 21, www.ubiq.com/weiser/calmtech/calmtech.htm, Acesso em: Jan/2007

VRContext. Walkinside – Visualization for the Industry. http://www.walkinside.com, Acesso em Jan/2007

Zhang, X.; Furnas, G. (2002) "Social interactions in multiscale CVEs". 4th International Conference on Collaborative Virtual Environments, CVE'02, ACM Press, pp 31-38

Capítulos 10

Jogos e Entretenimento com Realidade Virtual e Aumentada

Romero Tori, Ricardo Nakamura, João Luiz Bernardes Jr, Roberto Cezar Bianchini, Eduardo Costa Jacober, Daniel Calife e Alexandre Nascimento Tomoyose

Laboratório de Tecnologias Interativas (Interlab) – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo (USP)

{romero.tori, ricardo.nakamura, joão.bernardes, roberto.bianchini, eduardo.jacober, daniel.calife, alexandre.tomoyose}@poli.usp.br

Abstract

In this chapter the evolution of computer games and its industry, the major related concepts, and the relationship among videogame, Virtual Reality and Augmented Reality technologies will be discussed. Game engines, their use in computer game and virtual environments development, and the use of tangible user interfaces are among other important novel technologies that will be analyzed here.

Resumo

Este capítulo discute evolução e mercado dos jogos de computador e principais conceitos envolvidos, relacionando essa tecnologia

com as de realidade virtual e aumentada. Serão destacados os motores de jogos (game engines) e sua utilização no desenvolvimento de jogo e ambientes virtuais, bem como as interfaces tangíveis, importantes em sistemas de realidade virtual e aumentada.

10.1. Introdução

Jogos Eletrônicos (JE) e Realidade Virtual (RV) são áreas que têm sua origem no início da década de 1960, e que acabaram se desenvolvendo de forma independente. Algum tempo após o surgimento desses dois campos, a área de Realidade Aumentada (RA) também se juntou às outras duas e às muitas outras que estudam a forma como se pode representar e simular situações e ambientes em computadores. A Computação Gráfica (CG) é a principal tecnologia por trás de JE, RV e RA. Dessa forma a evolução da CG impacta as outras três áreas, assim como demandas provenientes de cada uma delas influenciam a indústria e pesquisas de CG. Devido ao grande avanço ocorrido no *hardware* gráfico e na diminuição do seu custo, pesquisas que englobam JE, RV e RA tendem a produzir resultados cada vez mais interessantes que aumentam o grau de interatividade homem-computador. A gigantesca indústria de jogos impulsiona a criação de ferramentas, dispositivos de interação e *game engines* que podem ser aproveitados também para o desenvolvimento de protótipos e/ou aplicações finais de RV e RA. Dessa forma, conhecer a tecnologia de desenvolvimento de jogos é interessante para profissionais, estudantes e pesquisadores da área de RV e RA, mesmo que esses só pretendam desenvolver aplicações “sérias”.

Este capítulo apresenta conceitos e tecnologia de JE e analisa algumas de suas relações com RV e RA.

10.2. Conceitos

Esta seção trata da conceituação e discussão sobre a aplicação das tecnologias de JE, RV e RA no entretenimento digital, com maior ênfase na realidade aumentada, por se tratar de tecnologia mais recente e ainda pouco explorada pela indústria do entretenimento. Serão também discutidas semelhanças e diferenças entre técnicas e tecnologias de RV e JE.

10.2.1. Jogos Eletrônicos

Definir jogos eletrônicos é simples, ou seja, são jogos que utilizam dispositivos eletrônicos (como computadores, consoles de *videogame*, - inclusive os portáteis – e celulares ou PDAs) como seu meio, ou seja, para permitir a interação com ele, mostrar seu estado, armazenar suas regras etc.

Já a definição de jogo é mais problemática. Embora reconhecer intuitivamente se determinada atividade é ou não um jogo pareça tarefa simples, diversos autores vêm tentando fornecer definições que consideram adequadas, muitas vezes conflitantes entre si, sem que a comunidade de desenvolvimento e pesquisa de jogos tenha chegado a um consenso. No livro publicado dois anos após sua morte, o filósofo Ludwig Wittgenstein (1953) argumenta que uma única definição, baseada em seus elementos, não é suficiente para determinar o conceito de jogo, que deveria ser representado por um conjunto de definições. Uma das principais dificuldades está na diferenciação entre jogo, brincadeira, competição, entretenimento e atividades artísticas. Outra se refere às diferenças culturais e lingüísticas que tratam tais conceitos de forma diferente em diferentes sociedades. Por exemplo, em português "jogo" engloba três diferentes palavras da língua inglesa ("play", "game" e "gambling"). Por outro lado "play" possui um sentido mais amplo que o nosso "jogar". Para evitar dúvidas, quando nos referirmos a "jogo" neste texto estaremos tratando

desse sentido mais estrito, equivalente ao “game” da língua inglesa. Atualmente há também correntes que preferem excluir da classificação de jogos os programas de simulação (como o “The Sims”) e de histórias interativas. Do ponto de vista tecnológico, no entanto, há muitas similaridades entre essas aplicações e jogos eletrônicos, de forma que sob esse ponto de vista as tecnologias discutidas neste capítulo também se aplicam a esse tipo de produto de entretenimento.

A maioria dos autores concorda com alguns elementos que devem estar presentes em uma atividade para que seja considerada um jogo: interatividade, regras bem definidas, um ou mais objetivos a serem alcançados, obstáculos para alcançá-los e o entretenimento do jogador. Walther (2003) define jogo como transgressões que levam da realidade a um "modo de jogo", basicamente em função de limites sobre local, tempo e objetos e da existência de regras e estrutura não-convencionais.

Mesmo esses elementos são fonte de discordância. Os objetivos de um jogo, por exemplo, nem sempre são explícitos e fechados (*close-ended*), mas podem ser implícitos ou abertos (*open-ended*), ou seja, estão de alguma forma previstos nas regras mas são escolhidos pelo jogador conforme seu desejo. Por contar somente com objetivos *open-ended*, escolhidos pelo usuário e não impostos pelo jogo, muitos jogos não foram classificados como tal no passado. Os produtos da série "Sim" e boa parte dos MMOs contam principalmente com objetivos desse tipo. Chris Crawford (2003), em sua definição baseada em dicotomias, tenta separar jogo de quebra-cabeça e competição com base na natureza dos obstáculos apresentados ao jogador, mas atinge uma definição muito restrita para jogo que constantemente desafia a intuição. A finalidade da atividade é outro aspecto que gera dúvidas, já que atualmente é bastante comum o uso de jogos com objetivos além do entretenimento, dentre eles destacando-se o treinamento de

profissionais. Mesmo quando o entretenimento não é a principal finalidade de um jogo, aquele componente deve existir para caracterizá-lo como tal. É difícil concordar com Roger Caillois (1957) que inclui a não-produtividade como elemento de sua definição de jogo.

Se é complexo classificar atividades como jogo ou não, classificar os jogos eletrônicos em si também não é unanimidade. Uma classificação comum (mas nem sempre consistente) do mercado:

- *First Person Shooter* (FPS): jogos de ação em que o jogador tem visão em primeira pessoa e tem como objetivo eliminar adversários (controlados pelo computador ou por outros jogadores) utilizando diversos tipos de armas ou golpes. Não existe evolução do personagem controlado pelo jogador, exceto através da posse de itens que coleta ao longo do jogo;

- Estratégia: o jogador controla (mais ou menos diretamente) diversos personagens (normalmente chamados de unidades) e estruturas (usadas para criar ou melhorar unidades ou para defesa) que são criados dentro do jogo a partir de recursos limitados. A estratégia está no gerenciamento desses recursos, na escolha do que criar e no uso das unidades;

- *RPG*: A principal característica desses jogos é a possibilidade de evolução do personagem controlado pelo jogador (é menos comum que o jogador controle mais de um personagem, mas existem jogos em que controla grupos pequenos), cujas capacidades melhoram conforme se alcança algum objetivo. Essas capacidades normalmente são representadas numericamente por estatísticas como atributos físicos, sociais ou mentais, habilidades, "níveis" etc. É comum, mas não necessário, que a forma como essa evolução acontece, desde a criação do personagem, também esteja

sob controle do jogador, permitindo a sua customização (a forma como os personagens interagem com o jogo pode ser bem diferente dependendo de quais estatísticas foram privilegiadas). Parte da evolução normalmente é também a aquisição e uso de "equipamentos" melhores para o personagem, cuja disponibilidade normalmente é limitada pelo estágio de evolução dos mesmos;

- Simulação: jogos que simulam alguma atividade real, como o crescimento e gerenciamento de uma cidade ou a vida de um indivíduo. Alguns tipos específicos de simulação são tão populares que podem ser considerados como categorias separadas:

  - Esportes (normalmente esportes populares de equipe e esportes radicais como skate e snowboard);

  - Simuladores de vôo;

  - Corrida (quase sempre refere-se a corrida de automóveis e não é classificado como jogo de esporte).

- Ação: Jogos eletrônicos em que ação intensa (normalmente violenta) é o principal componente, sem grande preocupação com elementos como gerenciamento de recursos ou evolução do personagem, mas com mais obstáculos a vencer que os FPS (a mera locomoção no jogo e a aquisição de equipamentos podem ter obstáculos complexos, *puzzles* são comuns fazendo com que muitos desses jogos sejam classificados como *action-adventure*);

- Aventura: Jogos eletrônicos em que o principal obstáculo são *puzzles* e onde o elemento de ação é quase inexistente. Normalmente o desenrolar do jogo conta uma história (e essa história costuma ser um dos maiores atrativos desses jogos) bastante linear;

É interessante notar que um dos principais fatores para essa classificação parece ser o tipo de desafio ou obstáculo que o jogador deve enfrentar. São comuns classificações que são combinações dessas, como *Action-RPG* ou o já mencionado *Action-Adventure*. Outras classificações são feitas com relação à plataforma do jogo (computador, consoles específicos, jogos para celular...) ou à sua finalidade (jogos para treinamento, jogos educativos, *advergames* para divulgação de produtos ou conceitos etc.).

Outro conceito relacionado, e que raramente é tratado com rigor é o da "jogabilidade". Para o escopo deste texto, considera-se como sendo o análogo da usabilidade, no contexto dos jogos. Ou seja, como a usabilidade é a capacidade de um sistema computacional de atender às metas dos seus usuários, a jogabilidade é a capacidade do jogo eletrônico de atender às metas (normalmente de entretenimento) dos seus jogadores.

10.2.2. Game Engines

Em diferentes áreas da Computação, o termo "*engine*" é utilizado para denominar o núcleo de um sistema de software responsável por uma tarefa específica. Desta forma, existem *engines* de busca para bancos de dados, *engines* de inferência relacionados à inteligência artificial, entre outros. Na área de JE, existem divergências sobre o que define o que é um "*game engine*", ou seja um *engine* para jogos eletrônicos. Por exemplo, a abordagem de Eberly (2000) é limitada às técnicas de geração de imagens e gerenciamento de cenas. Para McShaffry (2003), existem diferentes tipos de *game engines*, tais como *engines* de renderização 3D e *engines* de simulação física. Lewis e Jacobson (2002) definem *engine* como uma coleção de módulos de código de simulação que não especificam diretamente o comportamento ou o ambiente do jogo. A definição adotada neste texto é um pouco mais abrangente:

um conjunto de sistemas de software parametrizáveis, integráveis e reutilizáveis, que fornecem serviços utilizados em um jogo eletrônico.

A principal motivação para o surgimento e proliferação dos *game engines* é a reutilização de código. A evolução dos computadores leva a um aumento na complexidade dos softwares desenvolvidos e isto também se aplica aos jogos eletrônicos. Como resultado, hoje em dia projetos de JE podem durar vários anos, consumindo milhões de dólares e envolvendo dezenas de pessoas. Neste contexto, os *game engines* surgem como ferramenta para reduzir o tempo de desenvolvimento e os riscos de projeto, ao fornecerem serviços fundamentais necessários para os jogos eletrônicos. Como ilustrado na figura 10.1, o *game engine* implementa os seus serviços sobre a plataforma computacional na qual o JE será executado. Sobre o *game engine* fica o JE que o utiliza, responsável por controlar a execução de seus serviços e fornecer os dados necessários, tais como modelos 3D e sons. É interessante observar que em geral, os *game engines* não abstraem muitos detalhes das plataformas de *hardware* em que são executados. Isso leva a um maior acoplamento entre o *game engine* e o *hardware*, mas é feito para não comprometer o desempenho do JE.
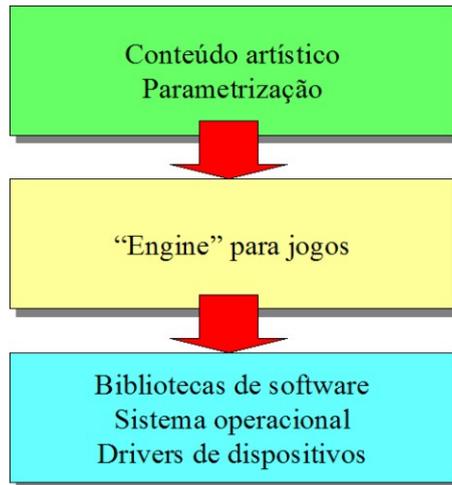
Figura 10.1. Estrutura de desenvolvimento de um jogo eletrônico que utiliza um *engine.*

A maioria dos *game engines* implementa algum tipo de geração de imagens, normalmente renderização de elementos tridimensionais. Isto se deve, principalmente, ao fato de que a maioria dos jogos eletrônicos atuais depende predominantemente da saída visual para interagir com o jogador. Uma conseqüência desta abordagem é que, em muitos *game engines*, a representação de entidades lógicas e visuais se confunde, sendo implementada de maneira fortemente acoplada.

Apesar da ênfase que se observa na capacidade de geração de imagens pelos *game engines*, existem muitos outros serviços que podem ser fornecidos por eles, tais como: geração de áudio, comunicação em rede, inteligência artificial, colisão e simulação física e tratamento de dispositivos de entrada.

Ao se utilizar *game engines* em aplicações de Realidade Virtual ou Aumentada, é preciso ter em mente o fato de que,

mesmo sendo flexíveis, esses sistemas de *software* são produzidos e otimizados para JE. Isto traz conseqüências como:

- Muitos *game engines* geram imagens adequadas para visualização em monitores convencionais. No caso de se utilizar dispositivos diferentes, tais como HMDs e CAVEs, é preciso avaliar a compatibilidade do *game engine;*

- A representação de objetos 3D e renderização é freqüentemente otimizada para determinados tipos de ambientes virtuais (cenários fechados ou terrenos amplos, por exemplo) e sua utilização em outros casos pode ser difícil ou mesmo inviável;

- Muitos *game engines* utilizam formatos próprios para modelos 3D, efeitos sonoros etc. e disponibilizam ferramentas de conversão entre formatos. No entanto, é preciso avaliar se os formatos suportados por tais ferramentas são adequados para o ambiente de produção específico (questões de licenciamento de *software*, tempo de processamento, conversões incompletas, etc);

- A tolerância à latência nos jogos eletrônicos tende a ser maior do que em ambientes virtuais multi-usuários, e os sistemas de comunicação em redes dos *game engines* são feitos de acordo com este fato. Portanto, pode ser necessário avaliar o desempenho do *game engine* caso seja utilizado para este tipo de aplicação;

- Os sistemas de colisão e simulação física utilizados nos *game engines* buscam gerar resultados “visualmente corretos”. Ao se utilizar tais sistemas em aplicações de RV e RA envolvendo simulação, deve-se avaliar se os resultados são adequados (fatores como confiabilidade e precisão).

Em resumo, os *game engines* possuem algumas características que os tornam semelhantes aos “*toolkits*” para

aplicações de Realidade Virtual. No entanto, sua implementação é direcionada para o desenvolvimento de JE e portanto seu uso em aplicações de RV e RA requer uma avaliação cuidadosa das características do *game engine*.

10.2.3. Interfaces Tangíveis

Influenciados pelo surgimento de duas novas tecnologias promissoras, a Realidade Aumentada e a Computação Ubíqua, Ishii e Ullmer (1997) definiram um novo tipo de interface homem-máquina muito mais ligada ao mundo físico (assim como RA e Computação Ubíqua): a Interface Tangível (TUI, *Tangible User Interface*). Esta interface se baseia no princípio da utilização de interações físicas para manipulação direta do mundo virtual, desta forma um objeto físico real e um objeto virtual são associados, quando ocorre alguma mudança em um deles (normalmente no objeto físico) o outro é afetado da mesma maneira, permitindo que o usuário manipule um objeto virtual em um ambiente 3D através da manipulação de um objeto real, de uma maneira muito mais intuitiva.

Através das interfaces tangíveis, a manipulação de conteúdos digitais se torna muito mais natural, como por exemplo rotacionar um cubo real e obter as mesmas transformações em um cubo virtual em um ambiente tridimensional. Um exemplo de aplicação da interface tangível é apresentado por Underkoffler e Ishii (1999), em uma aplicação chamada *"Illumination Light"*, nesta aplicação os usuários podem manipular objetos físicos que representam componentes ópticos (espelhos, *lasers*, lentes, etc) em uma "superfície aumentada". Quando o usuário posiciona o objeto que representa o *laser* nesta superfície, é projetado um raio luminoso que parece sair deste objeto, caso seja posicionado o objeto que representa o espelho, este raio será refletido de acordo com sua posição.

A colaboração entre os usuários é permitida e até mesmo incentivada com o uso das interfaces tangíveis, isto ocorre devido ao fato de as interações e movimentos terem menos limitadores, por exemplo, as interações com os marcadores físicos podem ser realizadas com as duas mãos e até mesmo com o corpo [Brave et al, 1998].

Para que o uso das interfaces tangíveis seja intuitivo e natural, sem a necessidade de um treinamento específico para sua utilização, é necessária a escolha de objetos físicos e metáforas comuns aos usuários que irão utilizá-las, assim os usuários podem basear-se nas suas habilidades e experiências desenvolvidas durante o curso de sua vida [Billinghurst et al, 2005].

Existem muitas aplicações e pesquisas de interfaces tangíveis aplicadas a JE [Bernardes et al, 2005; Paiva et al, 2003; Tangible Media Group, 2007]. Alguns jogos eletrônicos comerciais fazem uso deste tipo de interface, como o uso de pistolas, pedais, volantes, manches e até baquetas de bateria. Nem todas as aplicações de controles não-convencionais caracterizam-se como interface tangível, pelo simples fato de que nem sempre o controle é mapeado diretamente e unicamente com o elemento virtual correspondente. Assim, um controle na forma de cabo de raquete usado em um jogo de tênis virtual da mesma forma que um jogador desse esporte movimentaria uma raquete real é uma interface tangível. Já esse mesmo bastão usado para movimentar o curso na tela não seria uma interface tangível.

O console de *videogame* recentemente lançado “Wii” [Nintendo Wii, 2007], com seu controle sem fio com sensores de movimentos, pode atuar em alguns jogos eletrônicos como uma interface tangível, como por exemplo no jogo eletrônico *“Wii Sports”*, onde o controle pode ser utilizado como uma raquete de tênis ou como um taco de *baseball*.

10.2.4. Agentes Inteligentes e Vida Artificial

As aplicações em JE e RV são povoadas não apenas por seres humanos mas também por agentes artificiais, conhecidos em JE como *Non-Player Characters*, ou apenas NPCs. Estes agentes são objetos autônomos no ambiente, isto é possuem comportamento que não depende apenas da interação com os outros objetos para se manifestar, possuem a mesma representação no ambiente que os habitantes humanos (por avatares), mas seu comportamento é controlado por técnicas de Inteligência Artificial (IA). Dado a complexidade na geração das imagens gráficas em taxas de quadros por segundos adequadas, acima de 30 quadros por segundo, em JE e RV, durante muito tempo os algoritmos de IA utilizados foram muito simples, gerando comportamentos muito previsíveis e superficiais, não obstante pudessem ser divertidos em JE. Com o aumento no realismo gráfico, os usuários tanto de JE quanto de RV tendem a ter também uma expectativa alta em relação aos demais elementos que podem interagir no ambiente. Comportamentos muito artificiais e pobres para os agentes em JE e RV estão começando a quebrar o sentido de imersão dos usuários [Bianchini, 2005].

Além das técnicas sofisticadas para os comportamentos dos agentes, certas situações em JE e RV exigem comportamentos consistentes, que não quebrem a imersão, mas que são aplicados não a um, mas a um grupo de agentes. O importante não é o comportamento individual de cada agente, mas o seu comportamento global como um grupo. Neste sentido, técnicas de Vida Artificial (VA) trazem realismo para simulação de grupos, como um grupo de animais ou uma multidão de pessoas. Técnicas de VA também podem ser utilizadas na evolução do comportamento dos agentes. Resultados muito interessantes destas técnicas podem surgir devido ao seu caráter emergente: situações não previstas explicitamente inicialmente pelos desenvolvedores

podem emergir da interação dos diversos agentes entre si e com o meio, levando a experiências mais ricas e dinâmicas nestes ambientes.

10.3. Jogos e Realidade Virtual

Está cada vez mais freqüente o uso de JE como ferramentas ou objetos de pesquisa no meio científico [Rosenbloom, 2003]. Devido à grande popularização e demanda por poder de processamento visual, os jogos eletrônicos se tornaram o norteador do desenvolvimento de microcomputadores com alta capacidade gráfica.

Por serem explorados como produto de massa, os jogos eletrônicos ainda se limitam a plataformas computacionais de baixo custo. Por influência desta condição, os computadores pessoais que custam alguns milhares de dólares são capazes de renderizar cenas altamente complexas, custando dezenas, até mesmo centenas de vezes menos que uma estação gráfica utilizada em renderizações mais elaboradas, como em animações pra o cinema ou visualização científica de dados (Swartout and van Lent, 2003). As grandes empresas que produzem *hardware* gráfico visam primeiro antecipar e atender os requisitos dos estúdios desenvolvedores de JE, enquanto se contentam em atender às necessidades de visualização científica como um efeito colateral [Rhyne, 2002].

Uma vez tendo ajudado a popularizar o acesso a recursos gráficos em sistemas computacionais, a indústria de JE está influenciando o ambiente científico, sendo utilizados sistemas oriundos dos jogos eletrônicos (como os *game engines*) nas mais diversas aplicações, passando por propósitos militares [Kumagi, 2001], educativos e teatrais [Jacobson and Hwang, 2002]. Olanda descreve um sistema feito em RV com objetivos de entretenimento

que simula a exploração da superfície do planeta Marte [Olanda, 2006]. Em muitos casos *game engines* comerciais criados para JE servem de plataforma para a visualização de dados e simulações físicas.

Um outro uso resultante da melhoria gráfica dos jogos eletrônicos é a criação de novas formas de visualização de arte. Jacobson e Hwang (2002) adaptaram um *game engine* comercial para ser utilizado em uma CAVE ao invés de um monitor. Com o auxílio deste sistema é possível a exibição de mostras de arte de forma panorâmica, isto é, com um grande campo de visão.

Aplicações na área médica também são comuns. Slater et al (2001) descreve o uso de JE e RV no tratamento de fobias e traumas. Neste sistema, o paciente é exposto a um ambiente virtual, no qual são realizadas simulações de forma a auxiliar o paciente a superar o problema. A grande vantagem é que o paciente não está exposto à situação real, que poderia causar algum tipo de dano físico ou psicológico maior. Losh (2006) descreve um sistema de simulação em RV chamado *Virtual Iraq* que tenta diminuir os efeitos de estresse pós-traumático de soldados americanos após a invasão do Iraque por forças lideradas pelo EUA em 2003.

10.4. Jogos e Realidade Aumentada

Em sua primeira importante *review* sobre Realidade Aumentada, Azuma (1997) não mencionava jogos eletrônicos. Citava entretenimento como uma das aplicações dessa tecnologia, mas referindo-se principalmente à produção de filmes para o cinema. Já em 2002, jogos eletrônicos são citados por ele como uma importante aplicação, com diversos exemplos [Azuma, 2002]: uma mudança causada principalmente pelo barateamento do *hardware* usado em aplicações de RA, tornando-as apropriadas para o entretenimento doméstico. Treinamento também é uma importante

aplicação de RA citada por Azuma e muitos jogos têm sido usados para esse fim. Atualmente, até mesmo a indústria *mainstream* de jogos eletrônicos investe em RA, como evidenciado pelo "EyeToy" para Playstation2.

Uma preocupação constante da indústria de jogos atualmente, no entanto, como obter jogabilidade original em seus produtos [BusinessWeek, 2006], o que se torna cada vez mais difícil conforme os paradigmas com sucesso de jogos e interação são explorados incessantemente. Novas formas de interação, como o uso do controle do "Wii" como interface tangível e o já citado uso de RA com o "EyeToy", são soluções interessantes para esse problema, que exemplificam as vantagens que a RA pode trazer ao universo dos jogos eletrônicos. Outra vantagem é a possibilidade de combinar as características de jogos e brincadeiras clássicos (que não usam o meio eletrônico, como por exemplo jogos de cartas ou tabuleiro), como seu forte componente de socialização, com as dos jogos eletrônicos (gráficos e sons dinâmicos e atraentes, IA, comunicação em rede, manutenção das regras) através da combinação de elementos reais e virtuais [Magerkurth et al., 2004].

Por outro lado, jogos são uma aplicação bastante adequada para testar novas formas e dispositivos de interface, como RA [Starner et al., 2000]. Devido ao caráter lúdico e metáforas bem conhecidas em JE, os usuários se dispõem com mais facilidade a lidar com metáforas e tecnologias de interface mesmo quando ainda em estágio de protótipo e certamente com problemas. Além disso, a popularidade dos jogos auxilia na divulgação da nova tecnologia.

A principal classificação de jogos em realidade aumentada proposta nesse trabalho refere-se à área de jogo, separando os que fazem uso de grandes áreas, os que precisam de áreas limitadas e preparadas e os que se aproveitam da mobilidade de dispositivos

como celulares ou PDAs. As características, principalmente o tamanho, da área em que se passa um jogo em realidade aumentada têm grande influência não só em sua jogabilidade como na tecnologia usada em sua implementação.

Jogos projetados para áreas relativamente grandes (muitas vezes chamados “jogos *outdoor*"), como um campus universitário, ou um grande edifício, dependem de computadores “vestíveis” (normalmente carregados nas costas dos jogadores, em uma montagem especial similar a uma mochila). A saída gráfica é normalmente realizada através de HMDs , portáteis e mais viáveis do que distribuir monitores ou projetores em grandes áreas. Além disso, esses HMDs normalmente são do tipo semi-transparente, por razões de segurança. Um dos problemas mais complexos em realidade aumentada, o registro, torna-se ainda mais desafiador em grandes áreas e normalmente é necessária a combinação de várias técnicas de registro para tratá-lo. Um conhecimento prévio do ambiente, seja na forma de um modelo 3D ou planta em 2D, bem como o uso de GPS são constantes na implementação desses jogos eletrônicos, normalmente auxiliadas por técnicas de visão computacional ou sensores de movimento e orientação.

Em relação à jogabilidade, a metáfora física prevalece nos jogos para grandes áreas. Uma das maiores motivações para esses jogos é justamente trazer um jogo eletrônico para o mundo real e interagir de acordo. Sendo assim, para deslocar-se ou mudar seu ponto de vista em um desses jogos, o jogador anda, corre, pula ou vira a cabeça na realidade e assim por diante.

Já os jogos projetados para áreas menores e delimitadas podem fazer uso de diversas opções de tecnologia para sua implementação. Enquanto HMDs são comuns, a composição das imagens real e virtual por vídeo é tão comum quanto a ótica, ao contrário dos jogos para grandes áreas, e monitores e projetores em

diversas configurações também são largamente utilizados. Raramente há a necessidade de se usar computadores portáteis ou vestíveis, e a tarefa de registro pode ser feita de forma mais simples, utilizando-se uma única técnica isoladamente (normalmente visão computacional, muitas vezes com câmeras em posições fixas e conhecidas, mas sensores de posição e outras técnicas também são usados). Além disso, o jogo normalmente faz uso de algum elemento do ambiente, como uma mesa, caixas sobre uma grade desenhada no chão, uma parede de cor uniforme, um cenário etc.

Um terceiro tipo de jogo envolve dispositivos móveis e não depende do local onde esteja sendo jogado. Essa independência é que caracteriza esse tipo de jogo, e não somente o uso de dispositivos móveis, visto que os mesmos também podem ser usados em jogos eletrônicos em pequenas áreas. A principal dificuldade para esse tipo de aplicação é a capacidade de processamento consideravelmente mais baixa, atualmente, dos dispositivos portáteis.

Os principais jogos em RA para grandes áreas são o "*NetAttack*" [Fraunhofer, 2007], o "*Human Pacman*" [Cheok et al., 2004] e o "*ARQuake*" [Thomas et al., 2002]. Todos utilizam HMDs semi-transparentes, computadores vestíveis, GPS e comunicação por rede sem fio. O "*NetAttack*" e o "*Human Pacman*" utilizam mapas 2D do ambiente real, enquanto o "*ARQuake*" necessita de um modelo 3D. O "*Human Pacman*" é o único que não utiliza visão computacional e marcadores fiduciais para auxiliar no registro, utilizando sensores ativos. Cada um dos jogos tem peculiaridades interessantes, como o uso de uma "arma" com force feedback como dispositivo de entrada no "*ARQuake*", de interfaces tangíveis no "*Human Pacman*" e de colaboração com jogadores utilizando interfaces convencionais nele e no "*Net Attack*".

Parte da equipe responsável pelo desenvolvimento do "*ARQuake*", em conjunto com a iniciativa privada, está desenvolvendo um *hardware* e um *game engine* em "Realidade Aumentada" [A_RAGE, 2007]. A expressão está entre aspas pois, embora o "A_RAGE" combine elementos reais e virtuais através de um HMD semi-transparente e tenha interação em tempo real através de um *gamepad*, não há registro 3D entre os objetos reais e virtuais, não satisfazendo a definição de RA de Azuma (1996).

Jogos de RA em áreas limitadas, além de normalmente terem uma implementação mais simples e de menor custo, permitem uma maior variedade de metáforas. Talvez devido a esses fatores são bem mais numerosos que os para grandes áreas e serão apresentados de forma resumida na Tabela 10.1.

É muito comum que jogos sejam apresentados como sendo de RA, inclusive em trabalhos científicos, mesmo quando não combinam elementos reais e virtuais ou quando não fazem o registro 3D desses elementos, meramente por usarem HMDs ou marcadores fiduciais. Tais projetos não foram considerados aqui.

Nome/ referência	**Visualização/ composição**	**Registro**	**Interação**
Metáfora de Jogos de Tabuleiro			
KnightMage, CandyLand, Monopoly/ Magerkurth et al. (2004)	Monitores (Presentation Manager)/vídeo	Visão computacional (limitada), RFID, sensores magnéticos	Tangível (peças com RFID), PDA
Herding Sheep/ MacWilliams et al. (2003)	Projeção simples, monitor (laptop, PDA), HMD /vídeo e ótica	Visão computacional (ferramenta DTrack da ART)	Tangível, apontar e voz, laptop, PDA
Monkey Bridge /Barakonyi et al. (2005)	HMD/ótica	Visão computacional ou sensor magnético	Tangível (marcadores fiduciais ou "puck"

			magnético)
False prophets/ Mandryk et al. (2002)	Projeção simples/ótica	Diodos e sensores infravermelho	Tangível (peças com diodo infravermelho)
AR Mahjong/ Szalavári et al. (1998)	HMD/ótica	sensores magnéticos	Personal interaction panel (Szalavári e Gervautz, 1997)
Ulbricht e Schmalstieg (2003)	HMD/ótica	Visão computacional	Tangível (marcadores fiduciais)
Jumanji Singapore/ Zhou et al. (2004)	HMD/vídeo	Visão computacional	Tangível (cubos com marcas fiduciais)
Kanji Learning /Wagner e Barakonyi (2003)	Monitor (PDA)/vídeo	Visão computacional	Tangível (marcadores fiduciais)
TARBoard/Lee et al. (2005)	Monitor/Vídeo	Visão computacional	Tangível (cartas, tabuleiro)
MIND-Warping / Starner et al. (2000)	Projeção simples no tabuleiro ou HMD/ótica	Perceptive Workbench (visão e infravermelho) ou rádio	Tangível (peças) ou gestos e voz
Metáfora de Esportes			
AR^2Hockey/ Ohshima et al. (1998)	HMD/ótica	Rastreamento magnético da cabeça e visão computacional	Tangível (*mallets*)
Golf Simulator/ Govil et al. (2000)	HMD/vídeo	Visão computacional	Tangível (taco, bola)
AR Bowling/ Matysczok et al. (2004)	HMD/vídeo	Visão computacional, *dataglove* e sensor magnético	Não-tangível (bola)
AR Billiards/	HMD/vídeo	Visão computacional	Tangível (o próprio

Jebara et al. (1997)		(cor)	jogo de bilhar)
Ping Pong Plus/ Ishii et al. (1999)	Projeção simples/ótica	Som	Tangível (o próprio jogo de ping-pong)
CamBall/ Woodward et al. (2004)	Monitor/vídeo	Visão computacional	Tangível (raquete)
Metáfora Física			
RV Border Guards, AcquaGauntlet / Ohshima et al. (1999)	HMD/ótica	Sensores magnéticos e modelo 3D do ambiente	Gestos
Bladeships/ Takemura et al. (2004)	HMD/ótica	Sensores magnéticos (mão, cabeça) e modelo 3D do ambiente	Não-tangível (os bladeships)
Kick Ass Kung Fu/Hamalainen et al. (2005)	Projeção simples/vídeo	Visão computacional	Não-tangível (oponente)
Touch Space/ Cheok et al. (2002)	HMD/vídeo	Sensores magnéticos e visão computacional (marcadores fiduciais)	Gamepad, posição e co-localização, tangível (caixas)
AR PushPush/ Kim et al. (2005)	HMD/vídeo	Visão computacional (marcadores fiduciais)	Gestos
Outros			
ARHockey/ Vieira et al. (2006)	Projeção simples/ótica	Sensores infravermelhos	Tangível (batedores)
Câmera Kombat/ Paula et al. (2006)	Monitor/vídeo	Visão computacional	Gestos (posição corporal)

Invisible Train/ Wagner et al. (2005)	Monitor (PDA)/vídeo	Visão computacional (marcadores fiduciais)	Touchscreen
Glass Xylophone /Kim et al (2004)	Projeção simples/vídeo	Visão computacional (infravermelho)	Tangível (baquetas)
Interactive Storytelling/ Charles et al. (2004)	Projeção simples/vídeo (cromaqui)	Visão computacional (features)	Gestos, não-tangível e interpretação de linguagem natural

Tabela 10.1 - Jogos em Áreas Limitadas

O “*SymBall*” [Hakkarainen e Woodward, 2005], uma versão móvel do “*CamBall*” [Woodward et al., 2004], e o “chute a gol *mobile*" [Paelke et al., 2004] são os principais representantes dos jogos independentes de local. Ambos utilizam a câmera do portátil como dispositivo de entrada, através de técnicas de visão computacional, e a sua tela como saída gráfica.

10.5. Tendências

Uma tendência tanto para JE quanto RV é o aparecimento de ambientes virtuais tridimensionais genéricos, acessíveis por microcomputadores padrão pela internet, como o “*Second Life*” [Second Life, 2006]. Tais ambientes genéricos não têm um objetivo específico, fica a cargo de seus inúmeros partipantes (“habitantes”) decidir se utilizarão sua infra-estrutura para fins de comunicação (como pontos de encontro, propaganda, etc), fins de entretenimento (como jogos, exibições, etc), comerciais (venda de conteúdo ou serviços dentro do ambiente) ou de aprendizado (museus virtuais, ensino a distância, etc). Assim como os *game engines*, podem ser adotados como base para as aplicações científicas de RV e, da mesma forma, essa prática deve ser analisada com cautela, devido às suas características e restrições.

No campo dos *game engines*, novas oportunidades aparecem com relação à democratização ao acesso à plataforma de consoles, através de SDKs públicos (como o "*XNA Game Studio Express*" da Microsoft) e o formato de distribuição por *digital downloads*. Ao mesmo tempo, os novos consoles "Nintendo DS" e "Wii", bem como a série "*EyeToy*" da Sony, criam uma ampla base instalada de dispositivos anteriormente não-convencionais de interface, como telas sensíveis ao toque, microfones, sensores de posição e movimento, *force-feedback*, conexão sem-fio, captura de imagens e visão computacional. Ou seja, tais dispositivos podem facilitar tanto seu uso em projetos de pesquisa de RV, quanto diminuir o distanciamento do grande público a esses novos estilos interativos.

Alguns exemplos foram citados, anteriormente neste texto, da atividade batizada como "*serious games*", ou seja, a utilização das ferramentas e métodos característicos do desenvolvimento de jogos para aplicações diferentes do entretenimento. De modo geral, essa tendência se apresenta cada vez mais forte e mostrando ainda um grande potencial de crescimento, exemplificado por resultados favoráveis em um número crescente de aplicações e áreas do conhecimento.

Peters e Itti (2006) apresentam pesquisas voltadas a prever o comportamento do usuário e a possibilitar a criação de agentes com um comportamento mais humano. Este é um exemplo de evolução conjunta que pode ser aplicada tanto aos jogos eletrônicos como em aplicações científicas. Neste aspecto, a tendência atual é aumentar a complexidade dos comportamentos fazendo uso de técnicas de aprendizagem, para que os agentes possam se adaptar melhor às diversas situações que surgem nestes ambientes; incluir conceitos de computação afetiva tanto no processo de inferência quanto na representação externa que os usuários têm acesso; técnicas que permitam aos agentes apresentar comportamentos

mais voltados a objetivos, como no modelo *Belief Desire Intention* de agentes; técnicas de Sistemas Multi Agentes para a cooperação entre os agentes na solução de problemas.

10.6. Conclusão

Este capítulo apresentou os principais conceitos relacionados à tecnologia de jogos eletrônicos, com destaque para *game engines* e interfaces tangíveis. Foi analisada a interação entre as áreas de RV e RA com a de jogos eletrônicos, tendo-se apresentado o estado da arte dos jogos baseados em tecnologia de RA. Finalmente foram discutidas algumas das principais tendências da indústria e pesquisa de jogos eletrônicos e suas relações com RV e RA. Para o leitor interessado em se aprofundar no assunto é apresentada a seguir uma vasta bibliografia, utilizada como referência para a elaboração deste texto.

10.7. Referências Bibliográficas

A_RAGE "A_RAGE: Augmented Reality Gaming Engine" [sítio web]. Disponível em: <http://www.a-rage.com>. Acesso em: março de 2007.

Azuma, R. "A Survey of Augmented Reality". In: Presence: Teleoperators and Virtual Environments, n. 6, p. 355-385, 1997.

Azuma, R.; Baillot, Y.; Behringer, R.; Feiner, S.; Julier, S. Macintyre, B. "Recent Advances in Augmented Reality". In: IEEE Computer Graphics and Applications, v. 21, n. 6, p. 34-47, 2001.

Barakonyi, I.; Markus W.; Thomas P.; Schmalstieg D. "MonkeyBridge: Autonomous Agents in Augmented Reality Games". In: ACM SIGCHI INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENTERTAINMENT TECHNOLOGY, 2005. Anais... ACM, 2005.

Bernardes Jr., J. L.; Dias, J. S.; Tori, R. (2005). “Exploring Mixed Reality User Interfaces for Eletronic Games”, In: Proceedings of WJOGOS 2005, 353-358.

Bianchini, R.C. “Uma Arquitetura Bdi para Comportamentos Interativos de Agentes em Jogos Computacionais”, [tese de doutorado], 2005.

Bianchini, R. C.; Bernardes Jr., J. L.; Cuzziol, M.; Jacober, E,; Nakamura, R.; Tori, R. (2006) “Jogos Eletrônicos e Realidade Virtual”, Fundamentos e Tecnologia de Realidade Virtual e Aumentada, R. Tori e C. Kirner (org.), Porto Alegre, SBC, p. 199-219. disponível em www.interlab.pcs.poli.usp.br

Billinghurst, M.; Grasset, R.; Looser, J. (2005). “Designing augmented reality interfaces”, ACM SIGGRAPH Computer Graphics, 39, 1, 17-22.

Brave, S.; Ishii, H.; Dahley, A. (1998). “Tangible interfaces for remote collaboration and communication”, In: Proceedings of the 1998 ACM conference on Computer supported cooperative work, 169-178.

Businessweek. “Electronic Arts: A Radical New Game Plan”. Disponível em: <http://www.businessweek.com/magazine/content/06_12/b3976086.htm>. Acesso em: março de 2006.

Caillois, R. “Les jeux et les homes”. Gallimard, 1957.

Charles, F.; Cavazza, M.; Mead, S.; Martin, O.; Nandi, A.; Marichal, X. “Compelling Experiences in Mixed Reality Interactive Storytelling”. In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 32-41.

Cheok, A. et al. “Human Pacman: a Sensing-Based Mobile Entertainment System with Ubiquitous Computing and Tangible Interaction”. In: WORKSHOP ON NETWORK AND SYSTEM SUPPORT FOR GAMES, 2, 2003. Anais... 2003. p. 106-117

Crawford, C. “Chris Crawford on Game Design”. New Riders, 2003.

Eberly, D. H., “3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics”, Morgan-Kaufmann, 2000.

Fraunhofer F. “NetAttack Homepage”. [sítio web] Disponível em: <http://www.fit.fraunhofer.de/projekte/netattack/index_en.xml?aspect=ar-gaming>. Acesso em: abril de 2006.

Govil, A.; You, S.; Neumann, U. “A Video-Based Augmented Reality Golf Simulator”. In: ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, 8, 2000. Anais... ACM, 2000.

Hakkarainen, M.; Woodward, C. “SymBall: Camera Driven Table Tennis for Mobile Phones”. In: ACM SIGCHI INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENTERTAINMENT TECHNOLOGY, 2005. Anais... 2005.

Hamalainen, P.; Ilmonen, T.; Hoysniemi, J.; Lindholm, M.; Nykanen, A. “Martial Arts in Artificial Reality”. In: INTERNATIONAL CONFERENCE FOR HUMAN-COMPUTER INTERACTION, 2005. Anais... 2005.

Hause, K. “What to Play Next: Gaming Forecast 1999-2003”. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003. Anais... 2003. p. 894-895.

Ishii, H.; Ullmer, B. (1997). “Tangible bits: towards seamless interfaces between people, bits and atoms”, In: Proceedings of the SIGCHI conference on Human factors in computing systems, 234-241.

Ishii, H. Et Al. “PingPongPlus: Design of an Athletic-Tangible Interface for Computer-Supported Cooperative Play”. In: ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1999. Anais... 1999.

Jacobson, J. and Hwang, Z. “UnrealTournament for Immersive Interactive Theater”. In: Communicatins of the ACM, 45(1):39–42, 2002.

Jebara, T.; Eyster, C.; Weaver, J.; Starner, T.; Pentland, A. “Stochasticks: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers”. In: INTERNATIONAL SYMPOSIUM ON WEARABLE COMPUTERS, 1, 1997. Anais... IEEE, 1997. p. 138-145.

Kim, I.; Lee, H.; Kim, H. “Magic Mirror: A new VR platform design and its applications”. In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 343-348.

Kim, K. et al. “ARPushPush: Augmented Reality Game in Indoor Environment”. In: INTERNATIONAL WORKSHOP ON PERVASIVE GAMING APPLICATIONS, 2005. Anais... 2005.

Kumagi, J. “Fighting in the streets”. In: IEEE Spectrum, 38(2):68–71, 2001.

Lee, W.; Woo, W.; Lee, J. “TARBoard: Tangible Augmented Reality System for Table-top Game Environment”. In: INTERNATIONAL WORKSHOP ON PERVASIVE GAMING APPLICATIONS, 2005. Anais... 2005.

Lewis, M., Jacobson, J. (2002), “Game Engines in Scientific Research”, In: Communications of the ACM, Vol. 45, No. 1, p. 27-31.

Losh, E. “Making Things Public: Democracy and Government Funded Videogames and Virtual Reality Simulations”, In: Sandbox Symposium July 29-30, 2006,

Macwilliams, A.; Sandor, C.; Wagner, M.; Bauer, M.; Klinker, G.; Bruegge, B. “Herding Sheep: Live System Development for Distributed Augmented Reality”. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, 2, 2003. Anais... ACM, 2003.

Magerkurth, C.; Engelke, T.; Memisoglu, M. “Augmenting the Virtual Domain with Physical and Social Elements”. In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 163-172.

Mandryk, R.; Maranan, D.; Inkpen, K. “False Prophets: Exploring Hybrid Board/Video Games”. In: CONFERENCE OF HUMAN FACTORS IN COMPUTING SYSTEMS, 2002. Anais... 2002. p. 640-641.

Matysczok, C.; Radkowski, R.; Berssenbruegge, J. “AR-Bowling: Immersive and Realistic Game Play in Real Environments Using Augmented Reality”. In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 269-274.

McShaffry, M., “Game Coding Complete”, Paraglyph Press, 2003.

Nintendo “Wii.Nintendo.com”. [sítio web] Disponível em: <http://wii.nintendo.com/home.html>, acessado em Fevereiro de 2007.

Olanda, R.; Pérez, M.; Morillo, P. et al. “Entertainment Virtual Reality System for Simulation of Spaceflights Over the Surface of the Planet Mars”, In, VRST 06, November 1-3, 2006.

Ohshima, T. et al. “AR2Hockey: A Case Study of Collaborative Augmented Reality”. In: VIRTUAL REALITY ANNUAL INTERNATIONAL SYMPOSIUM, 1998. Anais... IEEE, 1998. p. 268-275.

Ohshima, T.; Satoh, K.; Yamamoto, H.; Tamura, H. “RV-Border Guards: A Multi-player Entertainment in Mixed Reality Space”. In: IEEE AND ACM INTERNATIONAL WORKSHOP ON AUGMENTED REALITY, 2, 1999. Anais... ACM, 1999.

Paelke, V.; Reimann, C.; Stichling, D. “Foot-based mobile Interaction with Games”. In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 321-324.

Paiva, A.; Chaves, R.; Vala, M.; Bullock, A.; Andersson, G.; Hook, K. "Towards Tangibility in Gameplay: Building a Tangible Affective Interface for a Computer Game", In: Proceedings of International Conference on Multimodal Interfaces ICMI'03, 60-67. 2003.

Paula, L.; Bonini, R.; Miranda, F. "Camera Kombat – Interação Livre para Jogos", In: Anais SBGames 2006.

Peters, R. J.; Itti, L. "Computational mechanisms for gaze direction in interactive visual environments". In: ACM:27-29, Março 2006

Rhyne, T. "Computer Games and Scientific Visualization". In: Communications of the ACM, 45(7):40–44, 2002.

Rosenbloom, A. "A Game Expirience in Everyday Application". In: Communications of the ACM, 46(7):28–31, 2003.

Second Life "Second Life: Your World. Your Imagination". [sítio web] Disponível em: http://secondlife.com Acessado em Fevereiro, 2007.

Slater, M., Pertaub, D. P. and Baker, C. "Medicine Meets Virtual Reality 2001: Outer Space, Inner Space, Virtual Space, volume Studies in Health Technology Studies" In: IOS Press, Amsterdam, Netherlands, 2001.

Starner, T. et al. "Mind-Warping: Towards Creating a Compelling Collaborative Augmented Reality Game". In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, 5, 2000. Anais... ACM, 2000. p. 256-259.

Swartout, W.; van Lent, M. "Making Game of a System Design". In: Communications of the ACM, 46(7):32–39, 2003.

Szalavári, Z.; Eckstein, E.; Gervautz, M. "Collaborative Gaming in Augmented Reality". In: ACM SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, 1998. Anais... 1998. p. 195-204.

Takemura, M.; Haraguchi, S. ; Ohta, Y. “An Interactive Attraction in Mixed Reality: Bladeships”. In: INTERNATIONAL CONFERENCE ON VIRTUAL SYSTEMS AND MULTIMEDIA, 10, 2004. Anais... 2004.

Thomas, B.; Close, B.; Donoghue, J.; Squires, J.; De Bondi, P.; Piekarski, W. “First Person Indoor/Outdoor Augmented Reality Application: ARQuake”. In: Personal and Ubiquitous Computing, n. 6, p. 75-86, 2002.

TMG “MIT Media Lab - Tangible Media Group”. [sítio web] Disponível em: <http://tangible.media.mit.edu/>, acessado em Fevereiro de 2007.

Ulbricht, C.; Schmalstieg, D. “Tangible Augmented Reality for Computer Games”. In: THE IASTED INTERNATIONAL CONFERENCE ON VISUALIZATION, IMAGING AND IMAGE PROCESSING, 2003. Anais... IASTED, 2003. p. 950-954.

Underkoffler, J.; Ishii, H. (1998). “Illuminating light: an optical design tool with a luminous-tangible interface”, In: Proceedings of the SIGCHI conference on Human factors in computing systems, 542-549.

Vieira, B.; Theodoro, C.; Trias, L.; Miranda, F.; Tori, R. “ARHockey: Um jogo de Realidade Aumentada Baseada em Projetores”, In: Anais SBGames 2006.

Wagner, D.; Barakonyi, I. “Augmented Reality Kanji Learning”. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, 2, 2003. Anais... ACM, 2003.

Wagner, D.; Pintaric, T.; Ledermann, F. ; Schmalstieg, D. Towards “Massively Multi-User Augmented Reality on Handheld Devices”. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING, 3, 2005. Anais 2005.

Walther, B. K.; "Playing and Gaming: Reflections and Classifications". In: GAME STUDIES – THE INTERNATIONAL JOURNAL OF COMPUTER GAME RESEARCH, vol. 3, issue 1, May 2003. Disponível em: <http://www.gamestudies.org/0301/walther>. Acesso em Março, 2007.

Woodward, C.; Honkamaa, P.; Jäppinen, J.; Pyökkimies, E. "CamBall – Augmented Networked Table Tennis Played with Real Rackets". In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais... ACM, 2004. p. 275-276.

Wittgenstein, L. "Philosophical Investigations". 1953.

Zhou, Z.; Cheok, A.D.; Chan, T.; Li, Y. "Jumanji Singapore: An Interactive 3D Board Game Turning Hollywood Fantasy into Reality". In: ACM SIGCHI ADVANCES IN COMPUTER ENTERTAINMENT, 2004. Anais ACM, 2004. p. 362-363.

Capítulo 11

Aplicações Médicas usando Realidade Virtual e Realidade Aumentada

Fátima L. S. Nunes[1], Rosa M. E. M. Costa[2], Ana Cláudia M. T. G. Oliveira[1], Sérgio R. Delfino[1], Larissa Pavarini[1], Ildeberto A. Rodello[1], José Remo F. Brega[1] e Antônio C. Sementille[1]

[1] Programa de Pós-Graduação em Ciência da Computação
Centro Universitário Eurípides de Marília (UNIVEM)
Avenida Hygino Muzzi Filho, 529 - CEP 17525-901, Marília – SP, Brasil

{fatima,lpavarini,rodello,remo,semente}@univem.edu.br, {anatiessi, srdelfino}@gmail.com

[2] Universidade do Estado do Rio de Janeiro
IME – Dept de Informática e Ciência da Computação
Rua São Francisco Xavier 524- 6º Bl. B, CEP 20550-013
Rio de Janeiro - RJ - Brasil

rcosta@ime.uerj.br

Abstract

Virtual Reality (VR) and Augmented Reality (AR) application for health area are developed with the most different purposes. Independent of the application objectives, requirements for its building can exceed that ones requested for application in other

areas. This chapter presents a practical vision of the important requirements to building VR and AR applications for health area, motivating a reflection and offering subsidies for planning tools for those people who desire start develop or are already developing application in this knowledge area.

Resumo

As aplicações de Realidade Virtual (RV) e Realidade Aumentada (RA) para a área de saúde são desenvolvidas para as mais diversas finalidades. Independente da finalidade da aplicação, os requisitos para construí-la muitas vezes podem ir além daqueles exigidos para aplicações destinadas a outras áreas. Este capítulo apresenta uma visão prática dos requisitos importantes para construir aplicações de RV e RA para a área de saúde, motivando uma reflexão e oferecendo subsídios para o planejamento de ferramentas daqueles que desejam adentrar ou estão desenvolvendo aplicações neste campo de conhecimento.

11.1. Introdução

Os computadores e tecnologias de comunicação vêm sendo utilizados de maneira crescente em diferentes áreas do conhecimento, sendo que, na última década estes equipamentos se disseminaram fortemente na área médica.

A tecnologia de Realidade Virtual (RV) vem despertando grande interesse neste domínio, pois amplia as possibilidades de estudo e prática de variadas técnicas e procedimentos médicos. Por outro lado, a difusão das redes favorece as aplicações de telemedicina, onde gráficos de alta resolução, imagens e áudio podem ser transmitidos e integrados em diferentes tipos de sistemas.

A conjunção destes avanços impulsiona o crescimento do número de experimentos e pesquisas que, por sua vez, exploram sofisticadas técnicas de visualização, incorporando equipamentos específicos capazes de gerar nos seus usuários sensações semelhantes às do mundo real.

Este capítulo discute a utilização dos computadores na área médica, aponta a exploração da Realidade Virtual e Realidade Aumentada (RA) como tendências para esta área, descreve requisitos e equipamentos necessários para o desenvolvimento dos trabalhos neste domínio e destaca várias aplicações. Como as questões éticas associadas às novas tecnologias extrapolam as fronteiras da tecnologia e atingem a sociedade de maneira direta, este capítulo apresenta, ainda, os valores éticos envolvidos no uso das novas tecnologias na área medica.

11.2. Aplicações – características, requisitos e exemplos

As aplicações desenvolvidas para a área médica estão inseridas em um contexto que exige ambientes virtuais realísticos e respostas a ações que permitam ao usuário ter a sensação de estar vivenciando uma situação ou procedimento como se fosse, de fato, real. Logo, um dos aspectos fundamentais para a utilização de ambientes virtuais neste domínio é a necessidade de geração de sensações próximas às reais, ou seja, gerar presença em seus usuários.

A presença ou telepresença pode ser definida como a sensação que a pessoa tem de realmente estar dentro do Ambiente Virtual, ou seja, estar em um local diferente do que fisicamente se encontra [Baños, 2000]. A presença pode ser obtida através do estímulo dos sentidos humanos (tato, visão e audição).

O senso de presença é dependente não somente das qualidades físicas (resolução, realismo, interatividade, tempo de resposta), mas também o que o usuário traz como bagagem

psicológica. Cada pessoa reage de maneira diferente quando expostas a mesma situação real ou virtual [Baños, 2000].

Neste sentido, North *et al.* (1998) observam que a concentração nas atividades realizadas no ambiente virtual está diretamente relacionada com o nível de interação e a adequação dos equipamentos de entrada e saída. A seguir, são apresentados alguns dos equipamentos mais utilizados em aplicações de RV e RA na área médica.

11.2.1. Equipamentos de Entrada e Saída

11.2.1.1. Equipamentos de Entrada

Em geral, os rastreadores são equipamentos muito utilizados em aplicações médicas, pois fornecem posicionamento em tempo-real e orientação de movimento de objetos ou pessoas, explorando diferentes tecnologias.

Eles podem ser do tipo mecânico, que fica preso à cabeça do usuário dando-lhe pouca mobilidade; ultra-sônicos, que trabalham por meio da emissão de ondas sonoras para determinar a posição do usuário e dando-lhe liberdade de movimentos, mas tendo pouca precisão; luminosos, que capturam, com o uso de câmeras, pequenas luzes acopladas ao corpo do usuário, sendo de uso confortável, mas consumindo muito tempo de processamento dos sinais; óticos, que trabalham de forma contrária aos rastreadores luminosos, com a câmera ficando presa à cabeça do usuário e as luzes em pontos estratégicos do ambiente físico; entre outros.

Estes rastreadores podem estar acoplados a instrumentos de manipulação cirúrgica [Aras, 2007], permitindo ao usuário ter precisão no controle dos seus movimentos e do mesmo tempo, gerando resistência. Estes equipamentos são classificados no grupo de interfaces hápticas e são essenciais na geração de sensações

próximas das obtidas em uma simulação real. O Capítulo 8 apresenta os equipamentos hápticos de forma detalhada.

As luvas também têm sido utilizadas em experimentos onde é necessária a manipulação de objetos [Haluck, 2007]. Capturam os movimentos das mãos e dos dedos, fazendo com que as respostas do ambiente sejam compatíveis com estes movimentos. Existem vários tipos de luvas funcionando com diferentes mecanismos de captura dos movimentos: tinta condutiva, esqueletos externos e medidores de luminosidade.

11.2.1.2. Equipamentos de Saída

Para a geração de telepresença, os ambientes virtuais necessitam de visão estereoscópica, que possibilitam a visualização de imagens com um mais alto grau de realismo, fornecendo sensação de profundidade ao observar o AV.

Um dos equipamentos que geram visão estereoscópica mais utilizados na área médica é o HMD (*Head-mounted display*), também conhecido como capacete. Permite isolar o usuário do mundo real e projetar imagens que geram um alto grau de presença. Entretanto, está no grupo de equipamentos de saída que gera mais problemas de saúde em seus usuários, tais como náuseas e tonturas. Os HMDs são utilizados em variados tipos de aplicações médicas, desde tratamentos de fobias e problemas neuropsiquiátricos [Costa, 2004] até a visualização de imagens para procedimentos cirúrgicos, por meio de técnicas de RA [Reitinger, 2006].

As Mesas de Projeção integram duas telas verticais em formato de “L”. Os usuários têm possibilidades de visualização estereoscópica e de interação com os objetos da cena [Barco, 2007].

A Caverna ou *Cave* é uma pequena sala onde imagens são projetadas sobre paredes translúcidas. Existem sensores de posição que capturam os movimentos dos usuários e atualizam as imagens

projetadas. Em geral, possuem som 3D, gerando sensações bastante reais. Sua grande vantagem é permitir que várias pessoas compartilhem a mesma experiência [Diffusion, 2007].

11.3. Criação de objetos tridimensionais

Em aplicações médicas, objetos tridimensionais (3D) são usados para representar pacientes, equipamentos, ambientes (consultórios, salas de cirurgia) e órgãos humanos.

Ferramentas para modelagens 3D oferecem inúmeros recursos para a construção desses objetos sintéticos. No entanto, é necessária uma certa dose de arte para construí-los, tarefa nem sempre trivial para profissionais de Computação. Além disso, algumas aplicações, como treinamento de procedimentos, podem exigir precisão em tempo real. Nesses casos, o ideal é que os objetos 3D sejam modelados a partir de dados provenientes de imagens médicas reais. Uma solução para implementar objetos realísticos é a utilização de imagens geradas por meio de equipamentos médicos. A seguir é apresentada uma conceituação sobre algumas modalidades médicas que podem gerar imagens para a criação de um Ambiente Virtual (AV). Exemplos de métodos de utilização dessas imagens são apresentados na seção 11.9.

As modalidades de diagnóstico por imagens se estabeleceram há algum tempo, valendo-se de uma variedade de equipamentos de aquisição como Tomografia Computadorizada (TC), Ressonância Magnética Nuclear (RMN) e Ultra-Som (US).

Na TC a imagem deriva do tratamento informatizado dos dados obtidos numa série de projeções angulares de raios-X. De maneira simplificada, traduz uma secção transversal (uma "fatia") do corpo do indivíduo (Figura 11.1a) [Wagner, 2002].

A Ressonância Magnética Nuclear (RMN) é uma técnica que permite determinar propriedades de uma substância por meio do co-relacionamento da energia absorvida com a freqüência, na

faixa de megahertz (MHz) do espectro eletromagnético. Usa propriedades dos núcleos dos átomos ou íons para mapear os tecidos, sob a influência de um campo magnético, produzindo imagens de alta qualidade (Figura 11.1b) [Hennel, 1993].

A ultra-sonografia é um método diagnóstico que aproveita o eco produzido pelo som para ver, em tempo, real as sombras produzidas pelas estruturas e órgãos do organismo. As ondas sonoras, ao encontrarem um anteparo, batem e são refletidas, retornando ao seu ponto de origem. Se for medido o tempo dispendido desde a emissão até a recepção do sinal de retorno, pode-se determinar com boa precisão a distância que ele percorreu, sendo possível a tradução em uma escala de cinza, que formará a imagem dos órgãos internos (Figura 11.1c) [Handolin, 2007].
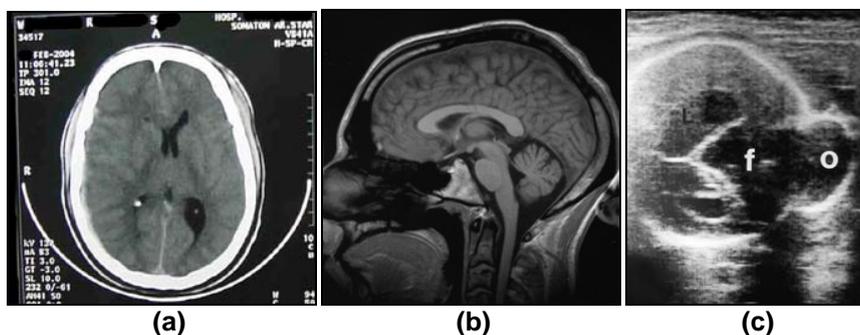
Figura 11.1 – Imagens de CT, RMN e US : (a) cérebro, obtida por CT; (b) cérebro, obtida por RMN; (c) cérebro de feto, obtida por US [Sayeg, 2007].

As modalidades que geram seções transversais (RMN, CT, entre outras) permitem um mapeamento direto para objetos virtuais 3D, por meio de técnicas de reconstrução de imagens. Para aquelas que geram apenas algumas visões, como US bidimensional (2D) e mamografia, é possível aplicar técnicas de simulação dos objetos 3D a partir de medidas extraídas das imagens 2D, como será visto na seção 11.9.

11.4. Detecção de colisão

Em aplicações médicas, precisão, tempo de resposta, conhecimento do local de impacto e verificação de interpenetração entre os objetos são fundamentais para a simulação de procedimentos de forma realística. Por isso, a detecção de colisão torna-se um fator extremamente relevante nesta classe de aplicações.

Lau *et al.* (2002) classificam este fator em dois grandes grupos: detecção em objetos rígidos, que envolve o teste de penetração de um objeto em outro e detecção em objetos deformáveis (superfícies flexíveis), que trata, além da interação entre os objetos, das deformações causadas por esta interação.

Na detecção de colisão em objetos rígidos, podem ser utilizadas três aproximações: subdivisão hierárquica do espaço, subdivisão hierárquica do objeto e cálculo incremental da distância [Lau, 2002].

Os métodos de colisão em objetos deformáveis, por sua vez, são baseados na subdivisão hierárquica do objeto, a qual pode ser de dois tipos: objetos representados por polígonos encadeados e por superfícies paramétricas (mapeamentos de algum subconjunto do plano ao espaço). No primeiro caso, os objetos são deformados alterando-se as posições dos vértices dos polígonos. Já os representados por superfícies paramétricas são, geralmente, deformados pela alteração do controle de pontos da superfície.

11.5. Deformação

Um AV pode ser composto de objetos estáticos, os quais mantêm a mesma forma durante toda sua existência e dinâmicos, que mudam de posição, forma ou aparência devido a alguma interação ocorrida. Para que um AV seja realístico, é necessário planejá-lo da forma mais próxima ao cotidiano, definindo suas ações e reações. A deformação é uma das reações a serem analisadas, sendo que o comportamento dos objetos deformados é representado por uma

simulação geométrica, física ou híbrida.

Choi *et al.* (2002) definem a simulação geométrica como aquela baseada em manipulações de dados geométricos como vértices e pontos. Usa artifícios matemáticos para obter resultados visualmente convincentes que permitem mudança na estrutura do objeto.

A simulação física utiliza a dinâmica, ramo da física que estuda as relações entre as forças e os movimentos que são produzidos por estas, ou seja, se preocupa com a causa e a conseqüência dos estímulos causados [Ramalho, 1993].

Na literatura três técnicas são as mais citadas para se obter deformação de objetos em aplicações de RV: *Free Form Deformation* (FFD) [Gibson, 1997], *Mass Spring* (MS), popularmente conhecido como Massa-Mola [Zill, 2003] e Elementos Finitos [Gladilin, 2003].

Free Form Deformation (FFD) é uma técnica geométrica que provê um alto nível de ajuste e controle individual dos pontos do objeto. O objeto é envolvido por uma grade que sofre deformação (Figura 11.2). Mesmo sendo versátil, Choi *et al.* (2002) afirmam que neste método é difícil limitar as deformações para pequenas regiões, dificultando a obtenção de realismo na cena quando empregado em um treinamento cirúrgico.

O Método dos Elementos Finitos originou-se da necessidade de resolver elasticidade complexa e análise de estruturas. Sua característica principal é a discretização da malha de um domínio contínuo em um conjunto de subdomínios discretos. A discretização do domínio é o primeiro passo em qualquer análise de elementos finitos, pois a maneira na qual o domínio é discretizado afetará as exigências de armazenamento no computador, o tempo de processamento e a precisão dos resultados numéricos [Hirota *et al.*, 1999].

A teoria da elasticidade afirma que a deformação de um corpo é proporcional à tensão aplicada desde que o limite elástico do material não seja excedido [Zill, 2003]. A análise de qualquer tipo de estrutura pode ser feita pela integração das equações da teoria da elasticidade. As dificuldades para realizar essa integração levam, porém, à formulação de modelos matemáticos aproximados. Os passos básicos para implementar este método podem ser obtidos em [Hirota *et al.*, 1999].
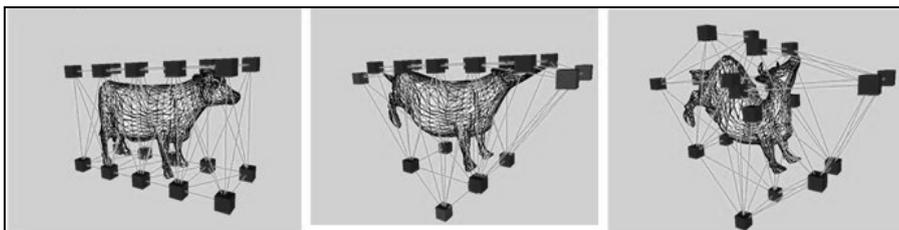
Figura 11.2 – Exemplo de objeto deformado utilizando o método FFD [Hirota *et al.* 1999].

O método Massa-Mola é uma técnica baseada na física que permite a remodelagem de objetos deformados através de nós de massa conectados por molas, como mostra a Figura 11.3 [Gibson, 1997]. Envolve formulação de matrizes e sistemas de equações diferenciais governadas por modelos dinâmicos para prevenir a instabilidade numérica. A mola pode ser considerada como uma estrutura que possui elasticidade e permite a deformação quando uma pressão é exercida nela. A força de uma mola é normalmente linear, baseada na lei de Hooke [Zill, 2003], mas molas não lineares podem ser usadas para simularem tecidos como pele humana, que exibem um comportamento não elástico.
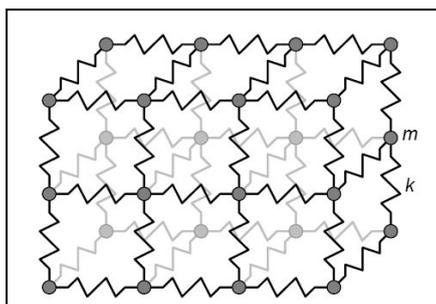
Figura 11.3 – Molas conectadas a um ponto de massa que exercem forças nos pontos vizinhos quando a massa é deslocada do resto das posições [Gibson, 1997].

11.6. Estereoscopia

Nas aplicações de RV e RA para proporcionar a sensação, denominada imersão, são empregados conceitos de estereoscopia.

A visão estéreo é um dos principais mecanismos que permite ao ser humano perceber a noção de profundidade. O olho esquerdo e o olho direito sempre vêem imagens diferentes (apesar de muito parecidas). Santos (2000) lembra que o cérebro usa esta diferença para montar a imagem. O princípio de funcionamento da maioria dos dispositivos estereoscópicos é a projeção de imagens distintas aos olhos esquerdo e direito do observador, proporcionando sensação de profundidade.

Há vários métodos para obter esta sensação. Johanson (2001) apresenta a técnica de vídeo estereoscópico, no qual posicionam-se câmeras separadas uma da outra que geram imagens distintas. O método de polarização da luz baseia-se na utilização de filtros, fazendo com que as imagens de um par estereoscópico sejam polarizadas em planos verticais e horizontais [Machado, 2003].

Óculos obturadores com lentes de cristal líquido são dispositivos especiais, cujas lentes ficam instantaneamente

transparentes ou opacas de acordo com um sistema de controle prévio, sincronizados com a imagem projetada. Existe também a visualização do par estéreo [Lipton, 1982], na qual são apresentadas duas imagens, lado a lado, de forma com que cada imagem seja posicionada levando em conta a distância entre os olhos do observador.

Ao método de estereoscopia que utiliza cores e diferentes profundidades codificadas, dá-se o nome de Disparidade Cromática. Esta técnica baseia-se na utilização de lentes especiais denominadas *ChromaDepth*, as quais realizam a codificação das profundidades através de suas cores. As lentes mudam a trajetória da luz que as atravessa de acordo com a cor do objeto, gerando o efeito estéreo.

O uso de anaglifos é apresentado por Schwartz (2004), explicando que os objetos são obtidos por uma combinação finita (azul-vermelho ou verde-vermelho) de cores, gerando uma sensação de profundidade e imersão no ambiente. Cada um dos olhos utiliza um filtro diferente de cor, dependendo da cor das lentes dos óculos estereoscópicos feito de papel celofane ou semelhante, para a visualização do par estéreo.

11.7. Precisão e tempo de resposta

Para que os simuladores sejam realísticos, é necessário que a resposta ocorra em tempo real e que os métodos implementados forneçam precisão.

Na Ciência da Computação, a expressão tempo real, em geral, refere-se à interação simultânea entre sistemas ou pessoas, com intervalos muito curtos ou com implicações sérias ao nível do atraso. É, portanto, uma operação em que o par ação/reação deve demorar menos tempo que o atraso máximo permitido pelo sistema [Shaw, 2003]. Essa conceituação de tempo real é importante em sistemas de treinamento médico que utilizam RV e RA, visto que

tais aplicações devem atender ao requisito de fornecer uma resposta dentro de um tempo máximo a fim de que usuário tenha a sensação de que a reação do sistema é imediata à ação que executou no AV.

Este fator não é obtido trivialmente, visto que modelagem e manipulação de objetos 3D, a interação por meio de dispositivos convencionais e não convencionais e o uso de bibliotecas gráficas contribuem para a inserção de atrasos ao sistema.

O fator precisão é outro requisito que merece destaque. Ao contrário de outros campos de aplicações, como indústria e entretenimento, a área médica, muitas vezes, trabalha com partes do corpo humano de tamanho diminuto. Em uma aplicação de telecirurgia, por exemplo, erros na ordem de milímetros são inadmissíveis. A necessidade de realismo exige que muitos sistemas, principalmente aqueles que simulam procedimentos, forneçam precisão em relação à modelagem e manipulação de objetos. E este requisito pode vir de encontro à necessidade de tempo real, citada nesta seção. Desta forma, tempo e precisão são dois fatores quase que conflitantes. Planejamentos adequados de aplicações devem prever uma relação coerente entre eles a fim de que a utilização dos sistemas desenvolvidos torne-se viável.

11.8. Exemplos de Aplicações

Inúmeros projetos de RV e RA estão sendo desenvolvidos com tecnologias e finalidades diversas, como pode ser verificado nos trabalhos destacados a seguir.

11.8.1. Educação e treinamento médico

Atualmente, a educação médica encara diversos desafios, seja no campo das ciências biológicas ou na área tecnológica e estão relacionados tanto à formação de futuros médicos, quanto à atualização de profissionais já formados.

A conjunção do rápido aumento dos novos resultados na área biomédica e da constante evolução das tecnologias associadas aos procedimentos clínicos, demanda uma constante atualização por parte dos profissionais da saúde. Para que o conhecimento adquirido não se torne rapidamente obsoleto, é imprescindível expandir e reforçar formas de educação permanente.

Por outro lado, o ensino da prática clínica requer contato direto entre estudantes de medicina e pacientes, com a supervisão de médicos e professores. Entretanto, os riscos de erros médicos envolvidos nesta prática são bastante significativos, podendo ocasionar problemas de saúde para os pacientes e problemas éticos e judiciais para os profissionais médicos.

Apesar da pouca difusão da educação apoiada em computadores nas escolas médicas, existem várias experiências sendo desenvolvidas. Estes sistemas podem simular casos clínicos em diferentes especialidades, diminuindo a necessidade de contatos reais entre médico e pacientes em fases iniciais do aprendizado médico.

Segundo Moline (1997) os ambientes virtuais provêem ferramentas educacionais que cobrem os aspectos experimentais através de uma abordagem didática, que explora principalmente, a visualização 3D de partes do corpo humano.

O estudo da anatomia em módulos virtuais diminui a necessidade do uso de cadáveres e oferece imagens com cores e comportamentos mais reais. Em um humano virtual é possível observar os órgãos em pleno funcionamento, diminuir os riscos de erros de secção e os efeitos psicológicos adversos gerados pelo contato com cadáveres reais.

Em geral, os Atlas para estudos da anatomia apresentam desenhos e fotos que não permitem uma visão mais integrada de órgãos e sistemas. Os Atlas digitais vêm suprindo algumas carências observadas nos Atlas em papel, entretanto, não é possível

a visualização estereoscópica, nem possibilidades de observação cooperativas.

O AnotomI [Monteiro, 2006] é um Atlas digital de uso livre e permite, de forma interativa, a manipulação e o estudo de estruturas tridimensionais do corpo humano com descritivos contendo informações textuais sobre cada estrutura. Este Atlas como recursos de RV e estereoscopia, imersão simultânea de várias pessoas, baixo custo de desenvolvimento e processamento e aplicação de transparências, entre outras. Permite, ainda, a consulta cooperativa que favorece o estudo e a discussão das estruturas estudadas.

O *Visible Human Project* [Visible, 2007] disponibiliza uma representação tridimensional de um corpo feminino e outro masculino, obtidas a partir da secção milimétrica de corpos humanos.

Ramos et al (2005) desenvolveram um Atlas virtual da mama para estudo da anatomia mamária e fisiopatologia do câncer de mama, através da visualização de estruturas, antes só disponíveis em imagens de livros ou estudo em cadáveres.

A RV aplicada à cirurgia abrange uma série de situações que vão desde o planejamento de um procedimento até o treinamento de técnicas e assistência para a sua realização. Neste sentido, o primeiro simulador cirúrgico desenvolvido no Brasil [Machado, 2003] visa a coleta de medula óssea. O sistema possui visualização esteroscópica e manipulação tridimensional de um modelo do osso ilíaco, dotado de camadas que possuem propriedades físico-elásticas, que são percebidas pelo usuário através da manipulação de dispositivos hápticos.

Na área de simulação de procedimentos merecem destaque a simulação de ortopedia de Sourin *et al.* (2000) e diagnóstico de câncer de próstrata de Burdea *et al.* (1999). Recentemente, Balanuik *et al.* (2006) desenvolveram uma ferramenta de

visualização de cirurgias plásticas dos seios, voltada para a educação do paciente e o planejamento cirúrgico.

11.8.2. Telecirurgia

A telecirurgia exige equipamentos sofisticados que possibilitam comunicação rápida e manipulação cirúrgica segura. Isto envolve o uso tanto de equipamentos de visualização 3D quanto de manipulação do campo cirúrgico, que disponibilizem ao cirurgião recursos para a realização dos procedimentos adequados, mesmo estando distante do paciente. No local aonde a cirurgia é realizada, é necessário um conjunto de itens que realizem a cirurgia no paciente. Neste caso os robôs são imprescindíveis, assim como uma rede que permita o envio e a recepção dos comandos de maneira rápida e segura [Current, 2007].

Estas tecnologias abrem novas possibilidades de tratamento para pessoas que se encontram distantes de grandes centros urbanos ou em situações de guerras.

Algumas dessas cirurgias podem utilizar os recursos da RA, que superimpõe imagens geradas a partir de exames de RMN ou TC em imagens reais.

11.8.3. Suporte a deficiências

Um outro tipo de aplicação médica da RA ajuda pessoas que possuem degenerações na retina (que geram manchas negras sobre as cenas) e requerem a integração de câmeras aos HMDs, permitindo que o usuário visualize imagens reais alteradas para suprir essa deficiência visual.

A Figura 11.4 apresenta um exemplo de aplicação onde as imagens reais (a) estão sofrendo um deslocamento em direção às bordas de uma área cega(b), permitindo que o usuário “enxergue” as imagens que estariam localizadas dentro dessa área (mancha negra) [Farago, 2005].

(a)

(b)

Figura 11.4 - Imagem original(a) e imagem transformada(b).

11.8.4. Reabilitação

A RV também tem sido explorada para apoiar o tratamento de diferentes seqüelas motoras e cognitivas, derivadas de distúrbios ou danos cerebrais. Em geral, os ambientes virtuais possibilitam uma variedade de associações não possíveis com outras interfaces homem-máquina, devido às qualidades multisensoriais e espaciais destes ambientes, contribuindo para o enriquecimento das aplicações na área de reabilitação.

Se uma pessoa sofre algum dano cerebral, uma ou várias funções motoras ou cognitivas podem se tornar deficientes. Para recuperá-las é preciso empreender estratégias terapêuticas específicas para cada tipo de deficiência detectada.

As pessoas que sofreram um Acidente Vascular Cerebral (AVC), em geral, sofrem de seqüelas motoras que são facilmente observadas, mas também, funções cognitivas podem ser danificadas. Neste sentido, Holden et al. (2006) propõem um sistema voltado para a *telereabilitação*. O ambiente permite ao terapeuta conduzir sessões interativas para o tratamento de pessoas com problemas motores nas mãos e que estão em suas casas. Um grupo de pesquisadores da UFRJ e UERJ vêm desenvolvendo

experimentos com ambientes virtuais 3D para estimular a realização de atividades de vida diária de pessoas com seqüelas de atenção e percepção causadas por AVC. Os pacientes têm aceitado com muito entusiasmo o trabalho nos ambientes virtuais e os resultados obtidos têm apresentado uma significativa recuperação dos níveis de atenção e concentração nas atividades cotidianas [Cardoso, 2006].

Costa e Carvalho (2004) apresentam os resultados da aplicação de um programa de reabilitação cognitiva para pacientes com esquizofrenia e insuficiência mental. As atividades propostas contemplam tarefas que estimulam a atenção e a memória. No experimento, verificou-se que os pacientes tiveram uma boa aceitação do trabalho com o computador e demonstraram um bom nível de motivação para usar a máquina.

11.9. Reconstrução de modelos a partir de imagens reais

Reconstruir um objeto significa recuperar suas informações geométricas e topológicas e seus atributos a partir do conjunto de amostras que o representa. Esta reconstrução é feita por meio de algum tipo de interpolação sobre as amostras.

Existem vários métodos de reconstrução e visualização de objetos tridimensionais. Os principais deles podem ser classificados em duas categorias: baseados em volume e baseados em superfície. Em aplicações de RV e RA para a área médica, a fonte de dados são as imagens provenientes de modalidades, como as citadas na seção 11.3.

Os métodos baseados em volume percorrem o conjunto de dados, gerando uma imagem a partir de alguns parâmetros como densidade, opacidade, textura e campos de vetores normais. Tais parâmetros indicam como os dados internos do volume são combinados e processados para gerar a imagem final, através da interação com a luz. Um exemplo pode ser visualizado na Figura

11.5. Uma das principais vantagens da reconstrução de volumes é a qualidade das imagens geradas. Um dos grandes problemas desta categoria de reconstrução é a grande quantidade de dados a processar [Peixoto, 2000].
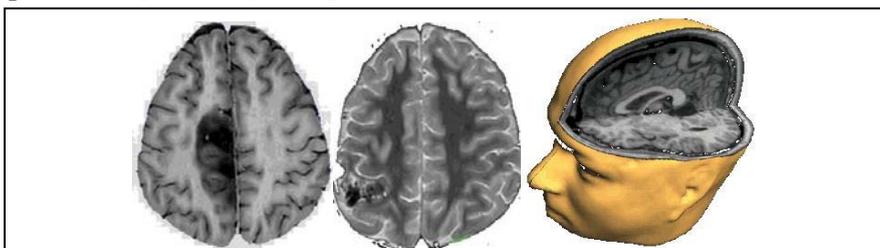
Figura 11.5 – Reconstrução volumétrica a partir de imagens de cérebro geradas por RMN (Neuroimage 2006].

A reconstrução da borda ou superfície que delimita um objeto sólido, a partir de uma série de seções planares paralelas, tem recebido bastante atenção nas últimas duas décadas [Peixoto, 2000]. A Figura 11.6 mostra um exemplo de contornos paralelos distribuídos em três fatias e a superfície reconstruída. Através de tecnologias como TC e RMN são obtidas informações sobre órgãos e tecidos humanos, armazenadas em fatias. As informações das fatias são utilizadas para definir o conjunto de contornos que, por sua vez, podem ser interpolados para gerar superfícies. Estas superfícies seriam as bordas dos órgãos ou tecidos humanos que, desta forma, podem ser exibidos e utilizados em aplicações gráficas. Na reconstrução por superfície dá-se importância somente às fronteiras que delimitam o objeto volumétrico. A partir de atributos como a função de densidade, é feita uma busca nas fatias do volume, a fim de definir o conjunto de contornos que delimitarão a superfície (exemplo na Figura 11.7). As técnicas desta categoria são utilizadas principalmente para reconstruir superfícies de objetos morfologicamente bem definidos. Uma das grandes vantagens dos métodos de reconstrução de superfícies em relação aos métodos de reconstrução volumétrica é a menor

quantidade de informações armazenadas. Uma vez construída a superfície, pode-se aplicar técnicas de renderização conhecidas para visualizar o objeto.
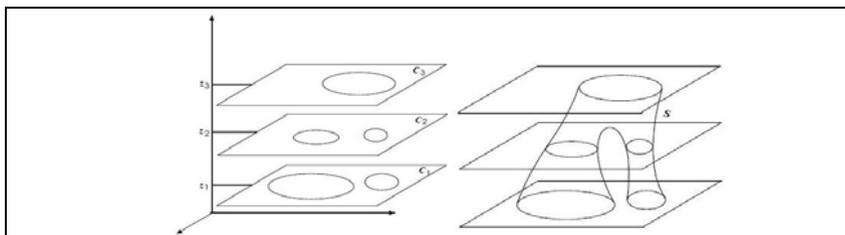
Figura 11.6 – Reconstrução de uma superfície a partir de contornos [Peixoto, 2000].

11.10. Detecção de colisão

Como introduzido na seção 11.4, a detecção de colisão com precisão é um fator relevante para proporcionar realismo em aplicações para a área médica. A seguir são citadas algumas soluções para alcançar este requisito.

O uso de hierarquias, juntamente com o conceito de volumes limites, é freqüentemente empregado para uma rápida detecção de colisão. Um volume limite representa a extensão espacial máxima de um objeto, podendo ser representada por um cubo ou esfera [Sense8, 2006]. Esta abordagem consiste em criar aproximações geométricas para objetos complexos, permitindo eliminar áreas dos objetos que não estão em contato.
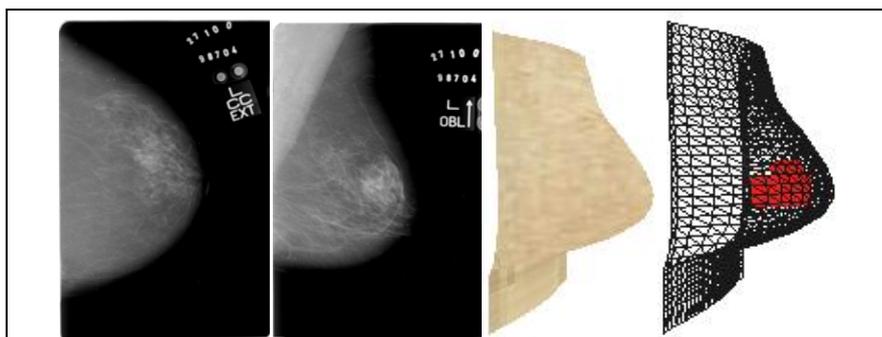
Figura 11.7 – Reconstrução por superfície a partir de imagens mamográficas: (a) e (b) imagens originais; (c) objeto com textura e (d) objeto *wireframe* [Delfino, 2006].

O'Sullivan *et al.* (2001) apresentam as técnicas mais comuns: (a) *Octrees*: hierarquias criadas recursivamente subdividindo o volume que contém um objeto em oito octantes (cubos ou esferas) e retendo somente aqueles octantes que contêm alguma parte do objeto original; (b) *Sphere-trees*: parecida com *Octrees*, esta técnica envolve os objetos por esferas, tendo como vantagem a rotação invariável; (c) *OBB-trees*: consistem no uso de volumes limites orientados com o objeto (*Oriented Bounding Boxes*) tendo como vantagem a proximidade com a forma do objeto e (d) *AABB-trees*: volumes limites alinhados aos eixos (*Axis Aligned Bounding Boxes*), consistindo em hierarquias mais simples de criar e realizar testes de sobreposição.

Especificamente na área médica, vários projetos encontrados na literatura buscam uma detecção de colisão precisa e com o menor custo computacional possível a partir de refinamentos ou combinação de métodos e algoritmos já existentes.

Em Garcia-Alonso *et al.* (1994) foram utilizados três métodos para determinar uma colisão entre dois objetos: Minimax,Voxels e Faces. Ponamgi *et al.* (1995) combinaram um algoritmo baseado no uso de volumes limites com aproximações

incrementais de suas características (lados, bordas e vértices) para detecção de colisão entre sólidos em ambientes dinâmicos.

Burdea *et al.* (1999) utilizaram o cálculo de distância entre vértices para a detecção de colisão entre a superfície de uma próstata virtual e um dedo virtual. Em [Machado, 2003], a opção adotada foi o refinamento das rotinas de detecção de colisão contidas na biblioteca háptica GHOST.

Riquelme (2005) realizou um estudo comparativo dos métodos oferecidos pelas bibliotecas *Java3D* e *WorldToolkit* [Sense8, 2006] para detecção de colisão. Observou-se que os métodos fornecidos por essas tecnologias não eram suficientes para uma detecção de colisão com precisão e rapidez e, por isso, propôs uma nova técnica que utiliza a equação do plano para refinar os métodos fornecidos.

11.11. Deformação

Como mencionado na seção 11.5, há métodos clássicos na literatura para simular deformação em tecidos flexíveis. O grande desafio continua sendo a relação precisão versus tempo de resposta.

Em aplicações médicas, o conceito de Elementos Finitos foi utilizado em Shi e Yan (2001), que desenvolveram técnicas para deformação de objetos e cortes em aplicações de cirurgia virtual, Costa e Balaniuk (2002), que se concentraram em métodos para deformar objetos em tempo real e Dimaio e Salcudean (2002), que apresentaram uma simulação de inserção de agulha virtual em tecidos flexíveis. Exemplos de utilização do método Massa-Mola são Delingette *et al.* (1999), que apresentam aplicações de simulações de cirurgias crânio-faciais e Oliveira *et al.* (2006), que implementaram um sistema em Java para simular a deformação em exames de biópsia de mama.

11.12. Estereoscopia

Em aplicações médicas, muitas vezes a implementação de estereoscopia esbarra no custo de equipamentos sofisticados que podem fornecer sensação de imersão mais adequada. A seguir são apresentadas algumas soluções para proporcionar esta sensação.

Meneses *et al.* (2002) desenvolveram um estudo para comparar as técnicas anaglífica e de polarização quanto à nitidez e realismo de um segmento neuroanatômico, concluindo que a melhor técnica em percentagem de eficácia foi a de polarização, apresentando melhor efeito de profundidade e maior nitidez. Óculos estereoscópicos obturadores foram usados para simular o treinamento de cirurgias ortopédicas [Sourin, 2000] e coleta de medula óssea [Machado, 2003].

Montanha e Nunes (2005) optaram por uma abordagem de baixo custo, desenvolvendo o método de anaglifos em Java, aplicado em um Atlas Virtual. Apesar de constituir uma solução barata, os autores observaram que métodos mais adequados devem ser aplicados em sistemas da área médica, pois os anaglifos causam distorção nas cores das imagens, prejudicando a sensação de realismo tão desejada neste tipo de sistema.

11.13. Construção de *frameworks*

Um *framework* é definido como um conjunto de classes abstratas e concretas usadas para o desenvolvimento de uma aplicação com domínio específico. *Framework* orientado a objetos representa uma categoria de artefatos de software potencialmente capazes de promover reúso de alta granularidade [Johnson, 1988]. Têm sido pesquisados devido a sua capacidade de alterabilidade, extensibilidade e, em longo prazo, rapidez no desenvolvimento de novas aplicações, diminuição de erros e aumento de qualidade do produto final.

Alguns *frameworks* na área de RV e RA aparecem na literatura, mas poucos deles são direcionados especificamente à

construção de aplicações na área médica. Bastos *et al.* (2005) afirmam que esses *frameworks* permitem que o usuário se concentre no desenvolvimento da aplicação, pois não é necessário se preocupar com a gerência do sistema de RV. Tais *frameworks* podem oferecer abstração de dispositivos e sistemas de projeção, grafos de cena especializados, interação com o AV, suporte a sistemas distribuídos e renderização distribuída. Dentre os frameworks direcionados à construção de aplicações genéricas de RV, é interessante destacar o Avango [Tramberend, 2001], o IVORY [Sprenger, 1998] e o ViRAL [Bastos, 2005].

Especificamente para a área médica, é interessante citar o *VPat (Virtual Patients)*, apresentado por Freitas *et al.* (2003) e desenvolvido com o propósito de suportar aplicações de computação gráfica na área médica e o *ViMeT* (*Virtual Medical Training)*, descrito por Oliveira *et al.* (2006), cujo conjunto de classes permite criar e manipular um ambiente de RV com o objetivo de simular exames de biópsia. O ViMeT tem como principal motivação a possibilidade de reúso de código que pode facilitar a construção de aplicações de RV para treinamento médico a partir do ViMeT. Também faz parte do projeto uma ferramenta de instanciação automática [Oliveira *et al.*, 2006].

11.14. A ética no uso da RV na área médica

O avanço dos preceitos de bioética tem gerado impactos em vários domínios das ciências médicas e perpassa, de maneira incisiva, o uso das novas tecnologias nesta área. Neste sentido, a utilização e a construção de artefatos tecnológicos devem receber especial atenção de maneira a não colocar em risco a saúde de seus usuários.

Várias dimensões devem ser consideradas na construção e uso destes artefatos e englobam aspectos tecnológicos e humanos. As questões tecnológicas estão centradas nos aspectos técnicos

relacionados, principalmente, ao desempenho do programa que controla procedimentos médicos, à definição e implementação dos meios de armazenamento de dados, aos meios de comunicação utilizados, à estrutura lógica, metodologias e padrões de qualidade do processo de desenvolvimento. As questões humanas tratam dos impactos dos programas nos seus usuários, verificam se o programa está resolvendo o problema almejado pelos usuários e se o está resolvendo de maneira correta, ou ainda, se está expondo o usuário a algum tipo de risco ou desconforto, dentre outros. O uso de ambientes virtuais 3D na área de saúde pressupõe o respeito aos pacientes, considerando estes fatores éticos.

No caso da tecnologia de RV, vários trabalhos focam nas questões envolvidas na imersão [Lewis, 1997]. Existem variados fatores que podem influenciar a geração de problemas, sendo que alguns são associados a aspectos técnicos do equipamento e construção do ambiente, enquanto outros, são inerentes ao próprio usuário. Pessoas de idade mais avançada podem ter problemas ao usar um capacete de projeção estereoscópica. A limitação do campo de visão gerada por estes equipamentos também pode contribuir para a ocorrência de sensações desagradáveis nos usuários.

Logo, precauções especiais devem ser tomadas para assegurar a segurança e o bem-estar de pacientes em ambiente virtuais 3D. Neste sentido, antes de expor os pacientes a estes ambientes virtuais, é fundamental que sejam verificadas suas características individuais, assegurando que eles não estejam sofrendo de infecções ou gripes; alcoolizados; sob o efeito de medicamentos que afetem as funções visuais ou perceptivas; sob o efeito de drogas; com histórico de desordens de equilíbrio ou com deficiências visuais.

11.15. Comentários Finais

Este capítulo apresentou uma visão da RV e da RA aplicadas à área da saúde, destacando os requisitos fundamentais para a construção e utilização de ambientes virtuais cada vez mais realísticos, exemplos de sistemas voltados para o apoio da educação médica, das telecirurgias e de tratamentos específicos, dentre outros.

A partir do que foi exposto, percebe-se o potencial da RV e RA para impulsionar o crescimento do número e da qualidade dos procedimentos médicos apoiados nestas novas tecnologias. Cabe destacar a necessidade de um trabalho cada vez mais interdisciplinar, integrando os diferentes especialistas envolvidos na concepção e uso destes produtos.

Desafios como custo da tecnologia de *hardware* e *software*, capacidade de processamento das máquinas, manipulação de grande volume de dados e familiarização dos usuários com as novas tecnologias ainda devem ser vencidos. Logo, uma vasta área de pesquisa se descortina, trazendo desafios tanto tecnológicos, quanto médicos e educacionais.

11.16. Referências Bibliográficas

ARAS - Augmented Reality Aided Surgery, em: http://www.vrvis.at/vr/aras/, visitado em janeiro de 2007.

Balaniuk, R.; Costa, I.; Melo, J.; (2006), "Cosmetic Breast Surgery Simulation", *VIII Symposium on Virtual Reality*, pp. 387-396.

Baños, R. M., Botella, C., Garcia-Palacios, A., Villa, H., Perpiña, C., Alcañiz, M. (2000), "Presence and Reality Judgment in Virtual Environments: A Unitary Construct?", *CyberPsychology & Behaviour* vol. 3, n. 3, pp. 327-335.

BARCO, Em: http://www.barco.com/VirtualReality/en/products/

product.asp?element=313, visitado em janeiro de 2007.

Bastos, T. A., Raposo, A. B., Gattas M. (2005), "Um *Framework* para o Desenvolvimento de Aplicações de Realidade Virtual Baseados em Componentes Gráficos", *Anais do XXV Congresso da Sociedade Brasileira de Computação*, pp. 213-223.

Burdea, G., Patounakis, G., Popescu, V., Weiss, R. E. (1999) "Virtual Reality-Based Training for Diagnosis of Prostate Cancer", *IEEE Transactions on Biomedical Engineering*, v. 46, n. 10, pp. 1253-1260.

Cardoso, L.; Costa, R. M.; Piovesana, A.; Costa, M.; Penna, L., (2006), "Using Virtual Environments for Stroke Rehabilitation". *IEEE- 5th International Workshop on Virtual Rehabilitation*, New York, pp. 1-5.

Choi, K.S *et al.* (2002), "Scalable Force Propagation Approach for Web – Based Deformable Simulation of Soft Tissues", *Web3D'02*, Arizona, ACM Proceedings, pp.185-193.

Costa, I. F., Balaniuk, R. (2001), "LEM - An Approach for Real Time Physically based Soft Tissue Simulation", *International Conference in Automation and robotics*, Seul–Korea do Sul.

Costa, R. M.; Carvalho, L., (2006), "The Acceptance of Virtual Reality Devices for Cognitive Rehabilitation: a report of positive results with schizophrenia", *Computer Methods and Programs in Biomedicine*, v. 73, n. 3, pp. 173-182.

Current Status of Telesurgery, em: http://www2.telemedtoday.com/articles/telesurgery.shtml, visitado em janeiro de 2007.

Delingette, H., Cotim, S., Ayache, N. (1999). "A Hibrid elastic

model allowingreal-time cutting, deformations and force-feedback for surgery training and simulation", In: Computer Animation 1999. Anais IEEE Computer Society, p.70-81.

Diffusion Tensor MRI Brain Visualization, em: http://www.nature.com/neuro/journal/v5/n11s/full/nn948.html, visitado em janeiro de 2007.

Dimaio, S.P., Salcudean, S.E. (2002), "Needle Insertion Modelling and Simulation", *Proceedings of The IEEE International Conference on Robotics and Automation*, pp.2098-2105.

DirectX, Site Oficial (2006), http://www.microsoft.com/windows/directx/default.mspx, visitado em novembro de 2006.

Farago, P.; Barros, L.; Landau, L.; Cunha, G.; Costa, R. M. (2005), "ATDV: An Image Transforming System". *Lecture Notes in Computer Science*, v. 3414, pp. 727-734.

Freitas, C. M. D. S., Manssour, I. H., Nedel, L. P., Gavião, J. K.,Paim, T. C., Maciel, Ânderson (2003) "*Framework* para Construção de Pacientes Virtuais: Uma aplicação em Laparoscopia Virtual", *Symposium On Virtual Reality- 2003*, Ribeirão Preto. v. 6, pp. 283-294.

Garcia-Alonso, A., Serrano, N., Flaquer, J. (1994), "Solving the Collision Detection Problem", *IEEE Computer Graphics and Applications*, v.14, n.3, pp. 36-43.

Gibson, S.F.F., Mirtch, B. (1997), "A Survey of Deformable Modeling in Computer Graphics", *Mitsubishi Electric Information Technology Center America*, pp.1-22.

Gladilin, E. (2003) "A Biomechanical Modeling of Soft Tissue and Facial Expressions for Craniofacial Surgery Planning", Tese de

Doutorado, FUBerlin.

Haluck R., Webster R., Wang W., LeFever A., Mohler B., (2007) "Tracking Laparoscopic Surgical Motions Using the 5DT DataGlove", em: http://cs.millersville.edu/~webster/haptics/handproject/handproject.html, visitado em janeiro de 2007.

Handolin, L., Partio, E. K. , Arnala, I., Pajarinen, J., Pätiälä, H., Rokkanen, P. (2007), "The effect of low-intensity pulsed ultrasound on bone healing in SR-PLLA rod fixed experimental distal femur osteotomy in rat", *J Mater Sci Mater Med*, February.

Hennel, J. W., Klinowski J. (1993), *Fundamentals of Nuclear Magnetic Resonance*, Longman Scientific & Technical, Essex - England.

Hirota, G., Maheshwari R., Lin M.C. (1999), "Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization", *SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, Michigan, pp.234 – 245.

Johanson, M. (2005), "Stereoscopic Video Transmission over the Internet", *Proceedings of WIAPP'01*, San Jose, Julho.

Johnson, R. E., Foote, B. (1988), "Designing reusable classes", *Journal of Object Oriented Programming*, v.1, n.2, pp.22-35.

Lau, R.W.H., Chan, O., Luk, M., Li, F. W. B. (2002), "A collision detection framework for deformable objects", *ACM VRST'02*, pp.113-120.

Lewis, C.; Griffin, M., (1997), "Human Factors Consideration in

Clinical Applications of Virtual Reality", *Virtual Reality in Neuro-Psycho-Physiology*; Ed. Giuseppe Riva, IOS Press.

Lipton, L. (2005), "Foundations of the Stereoscopic Cinema", Van Nostrand Reinhold Co. pp.77-79.

Machado, L. S. (2003), "A Realidade Virtual no modelamento e Simulação de Procedimentos Invasivos em Oncologia Pediátrica: Um estudo de caso no Transplante de Medula Óssea", Tese de Doutorado, Escola Politécnica de São Paulo, São Paulo.

Meneses, M.S. *et al.* (2002), "Estereoscopia aplicada à Neuroanatomia: estudo comparativo entre técnicas de filtro de cores e de polarização", *Directory of Open Access Journals*, vol.60, pp.769-774.

Moline, J., (1997), "Virtual Reality for Health Care: a Survey"; *Virtual Reality in Neuro-Psycho-Physiology*; Ed. Giuseppe Riva, IOS Press, Holanda.

Montanha, F., Nunes, F. L. S. (2005), "Construção de Atlas de Anatomia e Fisiopatologia do Câncer de Mama utilizando Realidade Virtual", *XVIII Sibgrapi - IV WTDCGPI*.

Monteiro, B.; Valdek, M.; Cunha, I.; Moraes, R.; Machado, L.; (2006), "AnotomI 3D: Um Atlas digital baseado em realidade Virtual para o ensino de medicina", *VIII Symposium on Virtual Reality*, pp 3-14.

Neuroimage, (2006), "Ressonância magnética de crânio convencional", Em: www.neuroimaging.com, Visitado em dezembro de 2006.

North, M.M., North, S. M., Coble, J. R. (1998), "Virtual Reality

Therapy: an effective treatment for phobias". *Virtual Environments in Clinical Psychology and Neuroscience*, Ios Press, Amsterdam.

O´Sullivan, C., Dingliana, J., Ganovelli, F., Bradshaw, G. (2001), "Collision Handling for Virtual Environments", *Eurographics Tutorial*.

Oliveira, A. C. M. T. G., Pavarini, L., Nunes, F. L. S., Botega, L. C., Justo, D. R., Bezerra, A**.** (2006), "Virtual Reality *Framework* for Medical Training: Implementation of a deformation class using Java", *ACM* SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry. Hong Kong –China.

Peixoto, A., Gattass, M. (2000) "Reconstrução de Superfícies a partir de Seções Bidimensionais", Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

Ponamgi, M., Manocha, D., Lin, M.C. (1995) "Incremental Algorithms for Collision Detection Between Solid Models". *ACM Siggraph*, Utah.

Ramalho Jr., F., Ferraro, N. G., Soares, P. A. T. (1993), *Os fundamentos da Física*, Editora Moderna.

Ramos, F.; Nunes, F.; Botega, L.; Damasceno, E.; Pavarini, L.; (2005), "Atlas Virtual da Mama e Fisiopatologia do Câncer de Mama utilizando Java3D". *Workshop de Aplicações de Realidade Virtual*, Uberlândia.

Reitinger B., Bornik A., Beichel R., Schmalstieg D., **(**2006), **"**Liver Surgery Planning Using Virtual Reality", *IEEE Computer Graphics and Applications*, Vol 26, n. 6, pp 36-47.

Riquelme, F. (2005), "Estudo comparativo de soluções para detecção de colisão em tecnologias de Realidade Virtual para o desenvolvimento de ferramentas para treinamento médico" Dissertação de Mestrado, Centro Universitário Eurípides de Marília, Marília.

Santos, E.T. (2000), "Uma Proposta para uso de Sistemas Estereoscópicos Modernos no Ensino de Geometria Descritiva e Desenho Técnico", *Graphica 2000*, Ouro Preto, pp. 1-8.

Sayeg, N. (2007), "Avaliação Morfofuncional", em: www.alzheimermed.com.br/m3.asp?cod_pagina=1058, visitado em janeiro de 2007.

Schwartz, J. (2004), "Special topics in CS: intro to Bioinformatics" 2004. http://www.settheory.com/bioinformatics_syllabus/ bioinformatics_syllabus.html>,Julho.

Sense8 (2006), "WorldToolKit", em: http://www.inition.co.uk/inition/product.php?URL_=product_software_sense8_worldtoolkit&SubCatID_=54>, visitado em dezembro de 2006.

Shi, J.Y, Yan, L.X. (2001), "Deformation and Cutting in Virtual Surgery", *Proceedings of the Intern. Workshop on Medical Imaging and Augmented Reality*, IEEE Computer Society.

Sourin, A. *et al*. (2000), "Virtual Orthopedic Surgery Training", *IEEE Computer Graphics and Applications*, vol.20, n.3, pp.6-9.

Sprenger, T. C., Gross, M., Bielser, D., Strasser, T., (1998), "IVORY - An Object-Oriented *Framework* for Physics-Based Information Visualization in Java", *Proceedings of the IEEE Symposium on Information Visualization*, IEEE CS Press, pp. 79-86.

The Visible Human Project, em: http://www.nlm.nih.gov/research/visible/visible_human.html, visitado em janeiro de 2007.

Tramberend, H. (2001), "Avango: A Distributed Virtual Reality *Framework*". *Proceedings of Afrigraph '01*, ACM.

Wagner, J. M., Noo, F., Clackdoyle, R., Bal, G, Christian, P. (2002), "Attenuation Correction for Rotating Slant-Hole (RSH) SPECT using Exact Rebinning", IEEE Transactions on Nuclear Science.

Zill, D.G. (2003), *Equações Diferenciais com aplicações em Modelagem*, Thomson.

Capítulo 12

Visualização de Informação com Realidade Virtual e Aumentada

Ezequiel Roberto Zorzal[1], Luciano Ferreira Silva[1], Alexandre Cardoso[1], Claudio Kirner[2] e Edgard Lamounier Jr.[1]

[1] Programa de Pós Graduação em Engenharia Elétrica – Universidade Federal de Uberlândia (UFU)
CEP – 38.400-902 – Uberlândia – MG – Brasil

ezorzal@gmail.com, luciano@mat.ufu.br, alexandre@ufu.br, lamounier@ufu.br

[2] Programa de Pós Graduação em Ciência da Computação - Universidade Metodista de Piracicaba (UNIMEP)
CEP – 13.400-911 – Piracicaba – SP – Brasil

ckirner@unimep.br

Abstract

Using Visualization of Information techniques, by computational resources, it is possible to convert numerical data and to show them visually by the use of images or other sensorial stimulus providing a better comprehension. Recent studies have shown that the three-dimensional visualization of information brings advantages for the people, being able efficiently to be carried through the use of Virtual and Augmented Reality environment.

There are a lot of developed applications making use of some of these methods. This chapter presents definitions, techniques, methodologies and the state of art of applications for three-dimensional visualization that use the technology of Virtual and Augmented Reality.

Resumo

Utilizando técnicas de Visualização de Informação, por meio de recursos computacionais, é possível converter dados numéricos e apresentá-los visualmente ao usuário por meio de imagens ou outros estímulos sensoriais, de forma que possam ser melhor compreendidos. Estudos recentes têm mostrado que a visualização tridimensional de informação traz vantagens para as pessoas, podendo ser eficientemente realizada utilizando ambientes de Realidade Virtual e Aumentada. Existem várias aplicações desenvolvidas que fazem uso de algum desses métodos. Este capítulo apresenta definições, técnicas, metodologias e o estado da arte de aplicações para visualização tridimensional que utilizam a tecnologia de Realidade Virtual e Aumentada.

12.1. Introdução

De acordo com a etimologia da palavra informação sua origem provém do latim *informationem*, “delinear, conceber idéia” [Houaiss and Villar, 2001]. O seu conceito carrega uma diversidade de significados que são aplicáveis a eventos que variam do nível cotidiano ao técnico. Por exemplo, informação pode ser definida como o estado de um sistema de interesse, a qual a mensagem é a informação materializada. Floridi (2005) define informação como qualquer tipo de padrão que influencia a formação ou transformação de outros padrões. Já Bekenstein (2003) enuncia informação com o resultado do processamento, manipulação e organização de dados de tal forma que represente um acréscimo ao

conhecimento da pessoa que a recebe. As palavras informação e dados são intercambiáveis em muitos contextos. Todavia, não são sinônimos. Por exemplo, de acordo com a observação de Gadomski (1993), dados referem-se a tudo que pode ser processado e informação refere-se a dados que descrevem um domínio físico ou abstrato.

No final do século XX, ocorreu a denominada explosão informacional, que caracteriza-se pela aceleração dos processos de produção e a disseminação da informação e do conhecimento. O aumento da demanda de informações ocorrera principalmente devido: ao desenvolvimento da tecnologia de comunicação; ao avanço dos equipamentos de aquisição de imagens, de dados e sinais; e ao surgimento de computadores que permitem simular sistemas cada vez mais complexos.

Esta sobrecarga de informações dificulta o processo de análise, compreensão e utilização de dados. Por exemplo, é plausível ressaltar que tomar uma decisão correta, em qualquer área do conhecimento, com uma enorme quantidade de dados e pouco tempo, quase sempre é uma tarefa difícil de realizar. O computador pode, em poucos segundos, recuperar informações que um ser humano demoraria muito tempo para fazer a mesma coisa. Contudo, muitas destas informações são irrelevantes para o usuário, ou as informações consideradas úteis podem simplesmente ser perdidas, devido ao usuário não conhecer o relacionamento entre os dados. Fatos como estes motivaram a pesquisa em diversas áreas de estudo, envolvendo novos paradigmas para melhorar a representação de informações.

Uma abordagem advinda dessas pesquisas consiste na aplicação de técnicas de Visualização de Informação. Estas técnicas estudam formas de transformar dados abstratos em imagens reais, de forma a facilitar o seu entendimento e/ou ajudar

na descoberta de novas informações contidas nestes dados [Nascimento, 2005]. O objetivo final deste processo é auxiliar no entendimento de um assunto, o qual, sem uma visualização, seria mais difícil de ser compreendido.

Neste contexto, a área de Visualização de Informação se apresenta como um campo de estudo de grande utilidade, uma vez que faz uso de técnicas que facilitam o entendimento de informações a partir de representações visuais de dados.

Estas representações podem ser distribuídas em três classes: unidimensional, bidimensional ou tridimensional, que são definidas de acordo com a dimensão do espaço, onde os elementos geométricos utilizados estejam situados [Freitas, 2001].

As aplicações que utilizam o espaço tridimensional podem causar um impacto visual e despertar o interesse em diversos tipos de usuários, não somente pela maneira que os dados são representados na interface gráfica, mas também pelas novas formas de interação.

A visualização tridimensional de informações pode ser realizada por meio de ambientes de Realidade Virtual e Realidade Aumentada. Estas formas de visualização podem facilitar a análise e a compreensão dos dados, já que os mesmos podem ser dispostos de maneira intuitiva e interativa.

O presente capítulo aborda detalhadamente aspectos importantes da Visualização de Informação, por meio de Realidade Virtual e Realidade Aumentada, e descreve conceitos e definições essenciais para a compreensão deste tema, além de demonstrar as principais técnicas de visualização de dados aplicados em ambientes virtuais. Este capítulo, ainda disserta sobre o estado da arte de aplicações que utilizam estas tecnologias na área de

Visualização de Informação, ressaltando suas características e metodologias de desenvolvimento.

12.2. Técnicas de Visualização de Informação

Existem diversas técnicas que apóiam a Visualização de Informação, de acordo com a classificação de Keim (2002), estas técnicas podem ser divididas em:

- Projeções 2D/3D convencionais: abragem um grande número de técnicas mais simples e amplamente utilizadas como plotagem em planos e espaços, gráficos de barras, gráficos de pizza, gráficos de linhas, etc;

- Técnicas baseadas em projeções geométricas: têm como princípio o mapeamento de dados multidimensionais para padrões bidimensionais através da utilização dos valores presentes na base de dados como parâmetros para a geração de formas geométricas. Estas formas devem ser tais que o conteúdo da informação representada possa ser percebido e analisado visualmente em suas propriedades gráficas, sendo que quanto maior o número de propriedades percebidas individualmente, maior será o número de atributos dos dados discriminados. Como exemplos, têm-se as Coordenadas Paralelas [Inselberg and Dimsdale, 1990], as *Star Coordinates* [Kandogan, 2000] e os *Scatter Plots* como descrito em [Ward, 1994];

- Técnicas baseadas em ícones: cada item de informação é representado como um ícone, cuja aparência deve ser familiar ao ser humano, para que os atributos das entidades gráficas possam ser prontamente associados aos itens de dados em análise. Segundo [Pickett and Grinstein. 1988], cor, forma e textura são características amplamente exploradas no *design* dos ícones, pois podem ser utilizadas simultaneamente sem perda de informação.

Como exemplos, têm-se as clássicas Faces de *Chernoff* [Chernoff, 1973], os *Star Glyphs* [Chambers, Cleveland et al., 1983] e as *Stick Figures* [Pickett and Grinstein, 1988];

- Técnicas orientadas a *pixels*: apresenta-se cada atributo de um dado multidimensional, através de *pixels* do dispositivo de exibição, e faz-se o uso de cores para representar os valores dos dados. Também se calcula um fator, denominado Fator de Relevância [Keim and Kriegel, 1994], baseado na ordem de apresentação dos elementos. Cada dimensão é apresentada em uma janela individual, onde os elementos são comparados em relação a um atributo específico. A visualização pode ser gerada sobre todos os elementos de dados ou sobre um subconjunto especificado;

- Técnicas hierárquicas: nesta abordagem, o espaço k-dimensional é subdividido e os subespaços resultantes são apresentados de forma hierárquica, como por exemplo, na técnica denominada *Dimensional Stacking* [LeBlanc, Ward et al., 1990] que apresenta bidimensionalmente as dimensões em sucessivos níveis hierárquicos.

Existe ainda a possibilidade de combinação entre estas técnicas, que permite o surgimento de novas técnicas denominas híbridas.

12.3. Técnicas de Interação

Um conhecido ditado popular diz que “uma imagem vale por mil palavras”, sendo assim, pressupõe-se que um sistema interativo, com riqueza de imagens vale por muito mais. Sobre esta perspectiva, a ciência da Visualização de Informação define suas diretivas, com o objetivo de facilitar a análise de dados complexos: a Visualização de Informação deve ser interativa.

A interação é capaz de potencializar enormemente o poder elucidativo de uma dada técnica de visualização. Interagindo dinamicamente, o usuário pode alterar a visualização de forma que suas metas de exploração possam ser alcançadas. Meios adequados de interação permitem ao usuário criar diversos arranjos de estrutura da base de dados que esta sendo explorada, comparar suas dimensões e gerar conhecimento, a partir da análise das projeções geradas em cada passo do processo.

Existem diversos recursos, técnicas de interação e várias classificações. Grinstein and Ward (2002) procuram delinear a natureza das técnicas de interação e não classificá-las de forma explícita. Eles discutem os fatores que podem ser utilizados para agrupar os dados, segundo propriedades comuns. De acordo com Grinstein and Ward (2002), o usuário, acessando os dados em uma cena de visualização, pode fazer uso da interação, através de recursos de navegação, isto é, alternando parâmetros gráficos que lhe permitem ver a imagem por diferentes ângulos, buscando um quadro mais revelador. O usuário pode também utilizar a interação por amostragem dos dados para reduzir as proporções do processo de análise que viria a ser realizado sobre um conjunto menor de informações. Há também a interação direta, através da qual é possível fazer consultas para fins específicos, que surjam durante o processo de análise. E, por fim, é apontada a interação associativa que permite o acesso relacionado dos dados em diferentes técnicas de visualização.

12.4. Visualização de Informação com Realidade Virtual

Existem diversas tecnologias aplicáveis à área de Visualização de Informação e a Realidade Virtual vem se destacando nesse sentido. Esta expressão possui diversas definições, devido à sua natureza interdisciplinar e a sua evolução. Segundo Kirner (1997), pode-se definir Realidade Virtual como uma forma das pessoas

visualizarem, manipularem e interagirem com computadores e dados extremamente complexos, na qual idéias como imersão, interação e envolvimento com o ambiente virtual são consideradas básicas e fundamentais. Uma das principais vantagens desta tecnologia é o envolvimento amplo de sentidos do ser humano na interação homem-máquina. Sendo assim, um dos seus principais diferenciais é o seu avançado modo de interação.

Abaixo, são descritos alguns exemplos da aplicação da Realidade Virtual na Visualização de Informação.

Ferramenta Colaborativa para Visualização Tridimensional de Dados

Desenvolvida pelo Centro Universitário do Pará, em conjunto a Universidade Federal do Pará, a Ferramenta Colaborativa para Visualização Tridimensional de Dados [Sousa Junior, 2006] tem como função criar um ambiente, em que os usuários possam compartilhar a visualização e trocar informações pela *Web*. A visualização dos dados é feita em um ambiente virtual tridimensional, no qual o usuário pode interagir com as representações por meio de *zoom*, rotação, translação, etc.

Ao passar o mouse sobre um determinado gráfico, é possível verificar os seus detalhes específicos. Além disso, existe a possibilidade de personalizar algumas características da visualização dos dados, como, por exemplo, pode-se escolher os atributos dos eixos e os valores de seus intervalos. Após a visualização construída, o usuário poderá remover ou adicionar dados e/ou atributos. Existem duas visões, uma compartilhada e outra individual, onde o usuário pode realizar as interações com o sistema.

Os usuários devem ser identificados por *login* e senha e a troca de informação entre eles é feita em modo texto. Todos os usuários têm permissões iguais, no ambiente compartilhado, e como não possuem nenhum mecanismo para o controle de concorrência, faz-se necessário o uso de um protocolo social, ou seja, os usuários devem conversar entre si e decidir quem vai alterar o ambiente compartilhado em determinado momento. O protótipo tem como objetivos: a) Permitir a análise dos dados simultaneamente por mais de um usuário; b) Eliminar a barreira geográfica quando necessário; c) Eliminar a necessidade de conhecimento de tecnologia de acesso a banco de dados; d) Facilitar a análise de grande quantidade de dados; e) Facilitar a colaboração entre usuários.

Nas visualizações geradas, foram utilizadas cores e formas geométricas diferentes para representar os dados e suas grandezas. As cores fornecem os diversos atributos e a forma está relacionada ao valor numérico do dado. Para saber o valor real de qualquer atributo, basta passar o mouse sobre o dado desejado. A Figura 12.1 apresenta a interface desta aplicação.
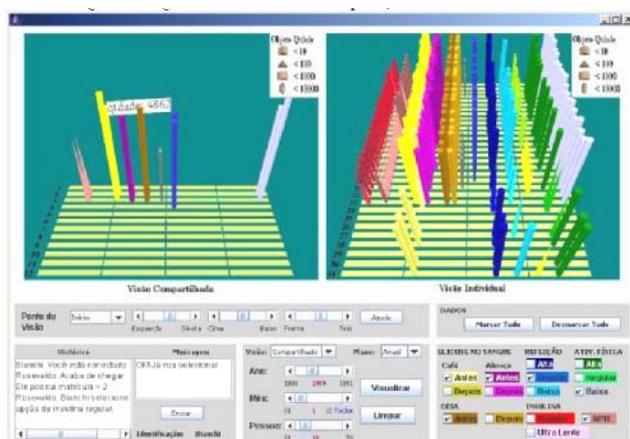
Figura 12.1. Interface do *software* [Sousa Junior, 2006].

InfoVis

Desenvolvido pela Universidade Federal de São Carlos, o *InfoVis* (*Information Visualizer*) [Martins, 2000] é uma ferramenta construída para visualizar informações, em um ambiente virtual, do Museu de Cerqueira César situado na cidade de São Carlos, Brasil.

O objetivo do sistema é gerar representações gráficas sobre o Museu de informações como: a freqüência de visitas diárias, semanais e mensais do Museu; tipos e características dos objetos armazenados; informações sobre os retratos e pinturas; etc.

O usuário poderá navegar no sistema e visualizar as informações do seu interesse por meio dos gráficos, em formato pizza ou barra, apresentados na tela. As cores dos gráficos são distintas a ponto de diferenciar os tipos de dados a serem representados. A Figura 12.2 ilustra uma das interfaces da aplicação.
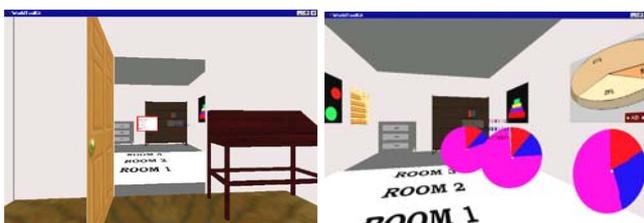
Figura 12.2. Ambiente do *InfoVis* (Martins, 2000).

3D Active Chart

A ferramenta *3D Active Chart*, desenvolvido pela *First Information Systems* [Systems, 2004], é um visualizador tridimensional de informações, no qual os dados são obtidos por meio de tabelas exportadas de um banco de dados.

As informações são apresentadas por gráficos construídos em VRML (*Virtual Reality Modeling Language*) [Ames, 1997] e pelos registros apresentados em uma grade. A cor das linhas dos registros apresentados nesta grade são as mesmas utilizadas nos gráficos. Ao clicar em um gráfico na cena, é possível visualizar seus valores correspondentes na grade. Além disso, pode-se ainda selecionar um ou mais registros na grade e visualizá-los como gráficos tridimensionais na cena.

É possível agrupar e filtrar os registros, de forma que facilite a manipulação e visualização. O usuário pode modificar o gráfico, alterando as informações que serão mostradas, a ordem de apresentação, a cor e o formato do gráfico. A Figura 12.3 apresenta a interface do *3D Active Chart*.
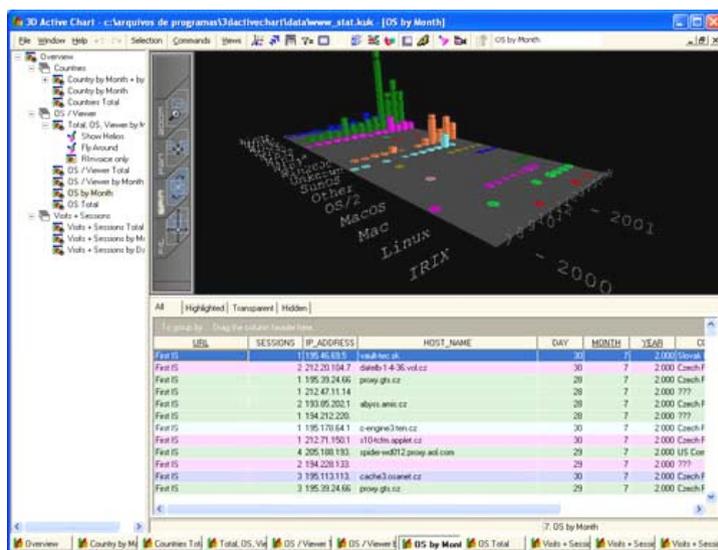
Figura 12.3. Interface do *3d Active Chart* [Systems, 2004].

12.5. Visualização de Informação com Realidade Aumentada

Para implementar interfaces tridimensionais, propiciando visualização e manipulação parecidas com as ações no mundo real, a Realidade Virtual faz uso de dispositivos especiais como luvas e capacetes. Isto se deve ao fato do usuário ter que entrar no contexto da aplicação executada dentro do computador, exigindo que ele conheça, ou seja, treinado no uso desses [Kirner and Tori, 2004]. Assim, apesar dos benefícios de uma interação mais natural, a necessidade de equipamentos especiais e de treinamento para seu uso acaba limitando o alcance da Realidade Virtual.

Uma solução para esse tipo de problema foi dada pela tecnologia de Realidade Aumentada que, misturando o cenário real com objetos virtuais gerados por computador, produz um único ambiente, sobreposto ao ambiente físico disposto na frente do usuário [Azuma, 1997], produzindo facilmente a análise e a interação com gráficos e a exploração de aspectos cognitivos, relatados com a compreensão da informação. Além disso, o usuário, utilizando as mãos, consegue manipular os objetos reais e virtuais do cenário misturado [Billinghurst, 2005], sem a necessidade de equipamentos especiais.

Abaixo são descritos alguns exemplos da aplicação da Realidade Aumentada na Visualização de Informação.

Visualização de Informação de Dados Multidimensionais Usando Realidade Aumentada

Foi desenvolvida pela Universidade Federal do Pará, uma aplicação para visualizar dados multidimensionais com Realidade Aumentada [Meiguins, 2006]. O sistema foi construído utilizando a biblioteca *ARToolKit* [Kato, 2000] como suporte, nesta aplicação o usuário pode interagir de forma natural e direta com a

representação, manipulando filtros e objetos tridimensionais. Cada objeto visualizado possui características específicas de cor, forma e tamanho, que representam diretamente valores dos atributos da base.

Por meio de obstrução dos marcadores, é possível acessar qualquer função do sistema. Esta ação consiste em interromper, por algum instante, o processo de captura da câmera sobre o marcador. Algumas destas funções consistem na utilização de filtros e transformações geométricas.

Os filtros implementados foram idealizados nos conceitos de consultas dinâmicas e contemplam tanto valores discretos, quanto contínuos. Os atributos discretos normalmente estão mapeados para características configuráveis de cor, forma e tamanho. Já os atributos contínuos podem ser representados nos eixos do gráfico. É possível também, interagir com o sistema por meio de mecanismos de *zoom*, rotação e translação.

Além das representações tridimensionais apresentadas pelo sistema, é possível visualizar gráficos 2D, que tem como objetivo auxiliar o usuário com novas informações sobre os dados que estão sendo visualizados. Existem dois tipos de gráficos bidimensionais, em formato pizza e no modelo de histograma. A Figura 12.4 elucida o ambiente da aplicação.
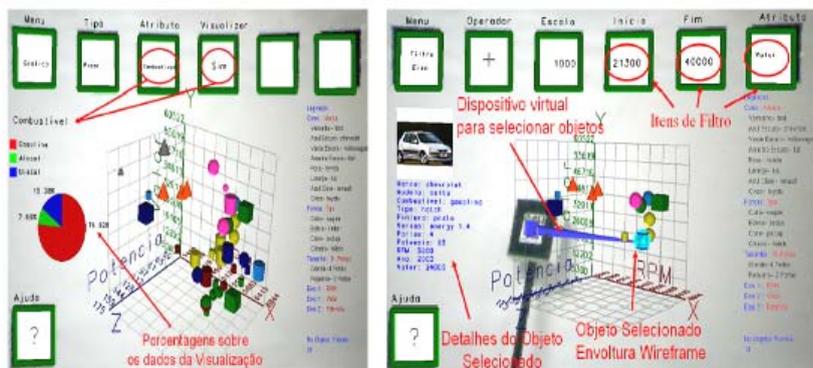
Figura 12.4. Interface do Sistema [Meiguins, 2006].

DataVis-AR

Desenvolvido pela Universidade Metodista de Piracicaba, o *DataVis-AR* (*Data Visualization Virtual Environment Using Augmented Reality*) [Buk, 2005] é um sistema interativo de visualização de dados com Realidade Aumentada, que utiliza como ferramenta de apoio o *software ARToolKit*. O *DataVis-AR* permite a análise de diferentes tipos de dados por meio de representações gráficas.

O *software* pode ser configurado, de acordo com a necessidade do usuário, além de possuir um módulo de controle que permite a manipulação de dados no mundo real. Para isto, basta apenas posicionar um marcador pré-cadastrado no campo de visão da câmera e visualizar as informações virtuais no ambiente. A Figura 12.5 apresenta o ambiente desta aplicação.
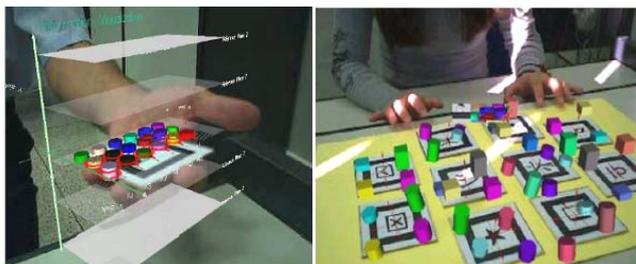
Figura 12.5. Interface do *DataVis-AR* [Buk, 2005].

Meta3D++

Desenvolvido pela Universidade Federal de Pernambuco, o *Meta3D++* é uma ferramenta de visualização de dados multidimensionais em ambiente de Realidade Aumentada [Bueno, 2005].

A ferramenta foi idealizada para que o usuário utilize em conjunto técnicas estatísticas e de Visualização de Informação, de uma forma integrada, agilizando o processo de criação de uma visualização mais efetiva. Um recurso disponível na *Meta3D++* é a categorização de variáveis, isto é, o usuário pode mapear as classes de uma tabela de distribuição de freqüências em categorias.

O sistema *Meta3D++* permite a criação de visualizações 3D, por meio de técnicas de faces de *Chernoff*, coordenadas paralelas e coordenadas paralelas extrusivas. O usuário pode construir as visualizações no sistema e exportá-las para o formato *VRML*, possibilitando uma fácil visualização e distribuição, Figura 12.6.
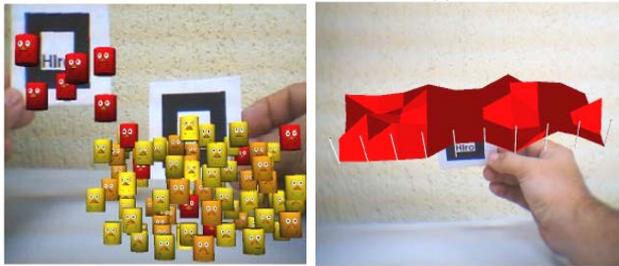
Figura 12.6. Interface do Meta3D++ [Bueno, 2005].

12.6. Considerações

A Realidade Virtual oferece muitos recursos para a visualização de informações, particularmente para a visualização de um grande volume de dados, uma vez que não limita o espaço de exploração dos dados a serem representados.

Observa-se, entretanto, que a interação com a Realidade Virtual requer a familiarização, treinamento e, eventualmente, o uso de dispositivos especiais para a navegação no ambiente tridimensional.

A Realidade Aumentada se apresenta como solução a tais problemas, uma vez que dispensa aprendizado e treinamento com as formas de interação tradicionais e/ou suportadas por Realidade Virtual.

Nos moldes do proposto com algumas tecnologias, encontra-se o problema relativo ao reconhecimento de marcadores (marcas fiduciais que fazem o mapeamento do objeto virtual para o cenário real), principalmente porque exige recursos de iluminação e posicionamento de câmera e/ou observador de forma adequada. Ademais, do ponto de vista da implementação e da concepção, fatores complicadores estão associados à dificuldade de programar os ambientes e as associações necessárias.

Ao se desenvolver um sistema tridimensional para visualização de informação devem ser considerados problemas como o da oclusão das mesmas. Em face disto, a mudança de visão e o método de navegação são essenciais para melhorar a interação e a percepção do usuário.

É importante ressaltar que nem sempre a melhor solução é o uso de ambientes tridimensionais para a visualização de dados. Às vezes, as informações a serem visualizadas são tão simples que uma apresentação com tais recursos torna-se fator complicador e prejudica, ao invés de melhorar, o bom entendimento das informações.

12.7. Referências Bibliográficas

Ames, L. A.; Nadeau, R.D.; Moreland D. VRML Sourcebook - Second Edition, John Wisley & Sons, Inc - USA, 1997.

Azuma, R. T. A Survey of Augmented Reality. UNC Chapel Hill, In Presence: Teleoperators and Virtual Environments 1997.

Bekenstein, J. D. Informação no universo holográfico. Scientific American 2003.

Billinghurst, M. et al. The MagicBook - Moving Seamlessly between Reality and Virtuality. Computer Graphics and Applications, 2001. 21(3), 2-4. Disponível em http://www.hitl.washington.edu/publications/r-2002-29/r-2002-29.pdf. Acesso em 30/03/2005.

Bueno, M. A. S., et al. Meta 3D++-Visualização de Informação em Realidade Aumentada. II Workshop sobre Realidade Aumentada. Unimep, Piracicaba, SP 2005.

Buk, C. V. Ambientes Virtuais para Visualização Dinâmica de Informação. 2005. Dissertação (Ciência da Computação) - Universidade Metodista de Piracicaba.

Chambers, J. M., Cleveland, W. S.; et al. (1983). Graphical Methods for Data Analysis. Belmont, CA, Wadsworth International Group.

Chernoff, H. (1973). "The Use of Faces to Represent Points in k-Dimensional Space Graphically." Journal of the American Statistica Association 68(342): 361 - 368.

Do Nascimento, H. A. D. Visualização de Informação - Uma Abordagem Prática. XXV Congresso da Sociedade Brasileira de Computação - XXIV JAI. UNISINOS - São Leopoldo - RS 2005.

Floridi, L. Is Information Meaningful Data? Philosophy and Phenomenological Research, 2005. 351-370 p.

Freitas, C. M. D. S., et al. Introdução à Visualização de Informação. Revista de Informática Teórica e Aplicada, v.2, p.143-158. 2001.

Gadomski, A. M. Patterns for Conceptualization of Knowledge on Technological Systems and Human Organizations. 9th Polish-Italian-Finnish Conference on Systems Analysis and Decision Support in Economics and Technology. Poland 1993.

Grinstein, G. G.; Ward, M. O. (2002). Introduction to Data Visualization. Information Visualization in Data Mining and Knowledge Discovery. U. Fayyad, G. G. Grinstein and A. Wierse, Morgan Kaufmann Publishers: 21-45.

Houaiss, A.; Villar, M. D. S. Dicionário Houaiss da Língua Portuguesa. Dicionário Houaiss da Língua Portuguesa. Objetiva. Rio de Janeiro - RJ: 2925 p. 2001.

Inselberg, A.; Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. IEEE Visualization, IEEE Computer Press.

Kandogan, E. (2000). Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions. IEEE Symposium on Information Visualization 2000, Salt Lake City, Utah.

Kato, H.; Billinghurst, M.; Poupyrev, I. ARToolKit version 2.33 Manual, Novembro. 2000.

Keim, D. A. (2002). "Information Visualization and Visual Data Mining." IEEE Transactions on Visualization and Computer Graphics 8(1):1-8.

Keim, D. A.; Kriegel, H. P. (1994). "VisDB: Database Exploration Using Multimensional Visualization." IEEE Computer Graphics and Application 14(5): 16-19.

Kirner, C.; Tori, R. Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade. In: (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. São Paulo - SP, 2004. Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade, p.3-20

Kirner, C. Virtual Environment for Shared Interactive Visualization. In: IV Workshop on Information Technology Cooperative Research with Industrial Partners between Germany and Brazil, 1997. Proceedings of the 4th Workshop on Information Technology Cooperative Research with Industrial Partners between Germany and Brazil. Porto Alegre - RS.

LeBlanc, J. ; Ward, M. O.; et al. (1990). Exploring N-Dimensional Databases. Proc. IEEE Visualization'90, San Francisco, CA.

Martins, V. F. Processo de Desenvolvimento de Ambientes e Aplicações de Realidade Virtual. 2000. Dissertação (Ciência da Computação) - Universidade Federal de São Carlos.

Meiguins, B. S. et al. Multidimensional Information Visualization Using Augmented Reality. VIII Symposium on Virtual Reality. Belém - PA 2006.

Pickett, R. M.; Grinstein, G. G. (1988). Iconographic Display for Visualizing Multimensional Data. IEEE Conf. on Systems, Man and Cybernetics, Piscataway, NJ, IEEE Press.

Sousa Junior, R. D., Meiguins, B. S.; Filho, M. R.. Ferramenta Colaborativa para Visualização Tridimensional de Dados. VIII Symposium on Virtual Reality. Belém - PA 2006.

Systems, F. I. 3D ACTIVE CHART. Disponível em: http://www.rinvoice.com/activechart.htm. Acesso em 28/02/2004.

Ward, M. O. (1994). XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. Proceedings of IEEE Conference on Visualization.

SEÇÃO D

AUTORES

Seção D

Autores e Editores

Robson Augusto Siscoutto (Organizador)

Editores do Livro

Claudio Kirner graduou-se em Engenharia Elétrica, em 1973, na Escola de Engenharia de São Carlos - USP (EESC/USP), concluiu o mestrado em Engenharia Eletrônica, em 1978, no Instituto Tecnológico da Aeronáutica (ITA) e o doutorado em Engenharia de Sistemas e Computação, em 1986, na Universidade Federal do Rio de Janeiro (UFRJ). Realizou Pós-doutorado na Universidade do Colorado em Colorado Springs (UCCS), Estados Unidos, no período de 1993 a 1995, desenvolvendo trabalhos sobre Realidade Virtual. Coordenou o I Workshop de Realidade Virtual, em 1997 e o I Workshop de Realidade Aumentada, em 2004 e outros subseqüentes. Foi coordenador da Comissão Especial de Realidade Virtual da SBC, no período de 2001 a 2004. Publicou cerca de 130 trabalhos científicos, 5 livros e 15 capítulos de livro. Orientou 7 doutorados e 22 mestrados e coordenou projetos de pesquisa financiados num montante superior a R$600K.
Áreas de interesse: Realidade Virtual e Aumentada, Ambientes Colaborativos e Educação à Distância.
Filiação: Programa de Pós-Graduação em Ciência da Computação, Universidade Metodista de Piracicaba (UNIMEP), e Centro Universitário Adventista de São Paulo (UNASP).
Contato: ckirner@unimep.br

Robson Augusto Siscoutto é professor titular da Faculdade de Ciência e Tecnologia da Universidade de Cuiabá – UNIC, onde coordena

o grupo de pesquisa em realidade virtual. Obteve grau de doutor na área de Engenharia da Computação e Sistemas Digitais pela Escola Politécnica da USP-SP em 2003. Membro do grupo de Tecnologias Interativas da USP-SP (INTERLAB). Outros graus acadêmicos são: Mestre em Ciência da Computação pela USP de São Carlos em 1997 e Bacharel em Ciência da Computação pela UNOESTE de Presidente Prudente em 1994. Publicou 3 livros e diversos artigos na área. Editor de dois livros na área de Realidade Virtual e Aumentada.
Áreas de Interesse: Realidade Virtual, Realidade Misturada, Tecnologias Interativas, Educação Virtual Interativa e Multimídia.
Filiação: INTERLAB - Laboratório de Tecnologias Interativas do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo. Grupo de Pesquisa em Realidade Virtual da Universidade de Cuiabá – MT. LAVI – Laboratório de Ambientes Virtuais Interativos da Universidade Federal do Mato Grosso – UFMT.
Contato: robson.siscoutto@poli.usp.br e robson@unic.br

Autores dos Capítulos

Alberto Raposo é coordenador de área do Tecgraf (Grupo de Tecnologia em Computação Gráfica), Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro. Obteve o grau de Doutor em Engenharia Elétrica, área de Engenharia da Computação pela Universidade Estadual de Campinas em 2000. Outros graus acadêmicos são Mestre em Engenharia Elétrica, área de Automação, UNICAMP, 1996; Engenheiro Eletricista, UNICAMP, 1994.
Área de Interesse: Realidade Virtual, Groupware e Ambientes Virtuais Colaborativos.
Filiação: Tecgraf (Grupo de Tecnologia em Computação Gráfica), Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro.
Contato: abraposo@tecgraf.puc-rio.br

Alexandre Cardoso é engenheiro eletricista pela Universidade Federal de Uberlândia (UFU) e doutor pela USP (POLI) na área de

Sistemas de Informação. Docente da Faculdade de Engenharia Elétrica da UFU, foi coordenador de curso e coordenador do Núcleo de Engenharia de Computação. Atualmente, é coordenador da Comissão Brasileira de Realidade Virtual (CERV) da SBC. Atuou como coordenador dos laboratórios de Informática e de Computação Gráfica e como assessor do Centro de Processamento de Dados. É co-autor e editor do livro "Ambientes Virtuais - Projeto e Implementação" (SBC, 2003) e do livro "Realidade Virtual - Uma Abordagem Prática" (Ed. Mania do Livro, 2004). Participa de projetos na área de Realidade Virtual (RV) e Realidade Aumentada (RA), entre eles "Lab. Virtual de Física" (Interlab USP, 2001), "RV no controle de consumo de Energia Elétrica", "Motor Elétrico Virtual", "Utilização de mapas conceituais e RV na educação (ensino médio e superior)", RA aplicada ao ensino e em jogos.
Áreas de Interesse: Educação Virtual Interativa, Tecnologias Interativas, Realidade Virtual, Realidade Aumentada.
Filiação: Grupo de Realidade Virtual da Faculdade de Engenharia Elétrica - Universidade Federal de Uberlândia (UFU)
Contato: alexandre@ufu.br - http://www.alexandre.eletrica.ufu.br

Alexandre N. Tomoyose é mestrando na Universidade de São Paulo (USP) e também pesquisador do Interlab (Laboratório de Tecnologias Interativas). Graduado em Engenharia de Computação pela Escola Politécnica (Poli-USP) com um projeto na área de engines de alto nível para jogos eletrônicos e trabalha com desenvolvimento web e jogos online desde então. Co-líder do projeto Robot ARena do Interlab, que consiste em um framework para reduzir o tempo de desenvolvimento de aplicações que implementem Realidade Aumentada Espacial.
Áreas de Interesse: Aplicações Web, Jogos Eletrônicos e Entretenimento Digital, Realidade Aumentada, Interface Homem-Computador.
Filiação: Interlab – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo.
Contato: alexandre.tomoyose@poli.usp.br e alextomoyose@gmail.com

Ana Cláudia Melo Tiessi Gomes de Oliveira é professora auxiliar da Faculdade de Tecnologia de Garça FATEC Garça. Obteve grau de mestre em Ciência da Computação na área de Realidade Virtual em 2007, no UNIVEM. Membro do LApIS – Laboratório de Aplicações

de Informática em Saúde. Outros graus acadêmicos são: Tecnóloga em Processamento de Dados pela Faculdade de Tecnologia de Ourinhos – FATEC Ourinhos em 2003.
Áreas de Interesse: Realidade Virtual, Treinamento Médico, Linguagens de Programação.
Filiação: LApIS - Laboratório de Aplicações de Informática em Saúde do Centro Universitário Eurípides de Marília (UNIVEM). FATEC Garça.
Contato: anatiessi@gmail.com

Antonio Carlos Sementille é Doutor em Física Computacional (IFSC/USP), Mestre em Ciência da Computação (UFSCAR) e Graduado em Ciência da Computação (UNESP Bauru). É pesquisador na área de Realidade Virtual desde 1995. Atualmente é docente no curso de Bacharelado em Ciência da Computação e no Programa de Pós-Graduação em Ciência da Computação do Centro Universitário Eurípides de Marília. É membro da Comissão Especial de Realidade Virtual (CERV) da Sociedade Brasileira de Computação.
Áreas de Interesse: Realidade Virtual, Realidade Misturada, Ambientes Virtuais Distribuídos e Compartilhados.
Filiação: LRV - Laboratório de Realidade Virtual - Centro Universitário Eurípides de Marília (UNIVEM).
Contato: semente@univem.edu.br

Bruno Barberi Gnecco é graduado em Engenharia Elétrica na Escola Politécnica da Universidade de São Paulo em 2002, onde adquiriu também o título de Mestre em Engenharia Elétrica em 2005. Participou do Núcleo de Realidade Virtual do Laboratório de Sistemas Integráveis da Universidade de São Paulo (LSI - USP), trabalhando em nas áreas de realidade virtual e computação distribuída. Atualmente trabalha em consultoria e desenvolvimento de tecnologias.
Áreas de interesse: realidade virtual, realidade aumentada, jogos, jogos on-line, sistemas distribuídos e de alto desempenho.
Filiação: Faculdade de Campo Limpo Paulista
Contato: brunobg@gmail.com

Daniel Calife é mestrando de engenharia elétrica (sistemas digitais) pela Escola Politécnica da Universidade de São Paulo e pesquisador do Laboratório de Tecnologias Interativas – Interlab, com interesse na aplicação de realidade aumentada em jogos e entretenimento digital. Bacharel em Ciência da Computação pela Universidade Cruzeiro do Sul.
Áreas de Interesse: Jogos e Entretenimento Digital, Realidade Virtual, Realidade Aumentada, Computação Gráfica e Tecnologias Interativas.
Filiação: Interlab – Laboratório de Tecnologias Interativas – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo.
Contato: daniel.calife@poli.usp.br

Denise Filippo é aluna de doutorado do Departamento de Informática da PUC - Rio desde 2004 e desenvolve tese sobre suporte computacional à coordenação em sistemas colaborativos. É mestre em Sistemas de Computação (1991) e engenheira eletrônica e de telecomunicações (1986), também pela PUC-Rio. Trabalha na área de projeto e gerência de redes na Universidade do Estado do Rio de Janeiro (UERJ) desde 1992.
Área de interesse: Suporte Computacional à Colaboração, Aprendizagem Colaborativa e m-learning
Filiação: Departamento de Informática da PUC-Rio e Instituto Politécnico - Campus Regional da Universidade do Estado do Rio de Janeiro em Nova Friburgo, RJ
Contato: denise@les.inf.puc-rio.br

Edgard Lamounier Jr. obteve o título de PhD em Ciência da Computação pela Universidade de Leeds-Inglaterra. Atualmente, ele é professor associado da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, onde desenvolve trabalhos de ensino, pesquisa e extensão. Ele já publicou diversos artigos nacionais e internacionais na área de Realidade Virtual e foi co-editor do livro Ambientes Virtuais Projeto e Implementação (SBC/SVR'2003). Além disso, ele é membro efetivo da Sociedade Brasileira de Computação e membro da Comissão Especial de Realidade Virtual da SBC.
Áreas de Interesse: Computação Gráfica, Realidade Virtual, Educação à Distância via Internet.

Filiação: Grupo de Realidade Virtual da Faculdade de Engenharia Elétrica - Universidade Federal de Uberlândia (UFU)
Contato: lamounier@ufu.br

Eduardo Costa Jacober é engenheiro formado pela USP, cursando mestrado pelo departamento de computação e sistemas digitais da Escola Politécnica da USP. Faz parte do grupo Interlab Games, voltado a pesquisas sobre jogos eletrônicos. Dentre os projetos do grupo, destaque para o game engine 3D "enJine" e o jogo "FootBot", desenvolvidos em Java, e na organização e participação de seminários e palestras sobre o assunto. Possui experiência profissional em projeto de redes de computadores, com certificações dos fabricantes Cisco e Nortel.
Áreas de Interesse: Jogos Eletrônicos, Interação Humano-Computador, Projetos Multidisciplinares, Criatividade.
Filiação: INTERLAB – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo
Contato: eduardo.jacober@poli.usp.br

Ezequiel Roberto Zorzal é professor contratado da Faculdade de Computação da Universidade Federal de Uberlândia (UFU). Graduado em Ciência da Computação pelo Centro Universitário Adventista de São Paulo (2005). Atualmente é dissente do curso de mestrado em Engenharia Elétrica da Universidade Federal de Uberlândia, atuando como pesquisador na área de Computação Gráfica, Realidade Virtual e Aumentada.
Áreas de Interesse: Realidade Virtual, Realidade Aumentada, Visualização de Informações.
Filiação: Grupo de Realidade Virtual – UFU – Pós-graduação em Engenharia Elétrica – UFU.
Contato: ezorzal@gmail.com – http://www.realidadeaumentada.com.br

Fátima L. S. Nunes é professora do Centro Universitário Eurípides de Marília (UNIVEM), onde coordena os Cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação. Fez pós-doutorado no Departamento de Engenharia Elétrica (EESC/USP) e Doutorado em Física Computacional (IFSC/USP). Coordena o

Laboratório de Aplicações de Informática em Saúde do Univem. Outros graus acadêmicos são: Mestre em Engenharia Elétrica (EESC/USP) e Bacharel em Ciência da Computação (UNESP Bauru). É pesquisadora nível 2 do CNPq.
Áreas de Interesse: Realidade Virtual, Processamento de Imagens, Banco de Dados, Engenharia Biomédica, Aplicações de Informática em Saúde.
Filiação: LApIS - Laboratório de Aplicações de Informática em Saúde - Centro Universitário Eurípides de Marília (UNIVEM) e LRV - Laboratório de Realidade Virtual - Centro Universitário Eurípides de Marília (UNIVEM).
Contato: fatima@univem.edu.br, fatima.nunes@pesquisador.cnpq.br

Hugo Fuks é professor associado do Departamento de Informática da PUC-Rio, Pesquisador 1D do CNPq e bolsista FAPERJ do programa Cientista do Nosso Estado - na categoria Pesquisador Inovador. Graduou-se em Engenharia de Produção pela UFRJ e obteve o doutorado em Computação pelo Imperial College London. Seu objetivo de pesquisa é dar suporte computacional à colaboração entre humanos. Na parte conceitual desenvolve uma Engenharia de Groupware baseada no modelo 3C de colaboração cujas dimensões são a comunicação, coordenação e cooperação, usando desenvolvimento baseado em componentes de programação. Na parte prática desenvolveu o AulaNet, um ambiente de ensino e aprendizagem na web, e mais recentemente vem desenvolvendo jogos didáticos, voltados para adolescentes em temas como educação financeira e ecologia.
Áreas de interesse: Suporte Computacional à Colaboração, Aprendizagem Colaborativa; Projeto de Interação
Filiação: Departamento de Informática da PUC-Rio
Contato: hugo@inf.puc-rio.br

Ildeberto Aparecido Rodello é Doutor em Física Computacional pelo Instituto de Física da Universidade de São Paulo (IFSC-USP), Campus São Carlos, Mestre em Ciência da Computação pela Universidade Federal de São Carlos (UFSCar) e Graduado em Tecnologia em Processamento de Dados pela Universidade Estadual de Campinas (Unicamp). É pesquisador na área de Realidade Virtual desde

1998. Atualmente é docente nos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação e no Programa de Pós-Graduação em Ciência da Computação do Centro Universitário Eurípides de Marilia (Univem), SP, sendo membro do Grupo de Realidade Virtual.
Áreas de Interesse: Realidade Virtual, Realidade Misturada, Ambientes Virtuais Distribuídos e Compartilhados, Realidade Virtual e Educação .
Filiação: Centro Universitário Eurípides de Marilia, UNIVEM, Marília, SP.
Contato: rodello@univem.edu.br

João Luiz Bernardes Jr. é engenheiro formado em 1998 pela Escola Politécnica da USP. Desde 1995 faz parte de projetos de pesquisa envolvendo desenvolvimento de software. Conclui em 2004 o mestrado na Escola Politécnica, englobando principalmente Visualização Científica e Realidade Virtual. Atualmente está no programa de Doutorado da mesma escola, desenvolvendo sua tese nas áreas de jogos eletrônicos e realidade aumentada. É um dos fundadores do grupo de pesquisa de jogos do Laboratório de Tecnologias Interativas da USP, onde tem participado de projetos (por exemplo, Jogo do Museu, enJine e FootBot), de pesquisas e aplicação de uso de jogos no ensino e da elaboração de artigos explorando tecnologias relacionadas a jogos eletrônicos, bem como ministrado palestras sobre o tema. Tem ainda experiência de 3 anos no ensino tanto para graduação como para pós-graduação.
Filiação: INTERLAB – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo
Contato: joao.bernardes@poli.usp.br

José Remo Ferreira Brega é Doutor em Engenharia Civil (EESC/USP), Mestre em Engenharia Civil (EESC/USP), Graduado em Engenharia Civil (EESC/USP) e Tecnologia em Processamento de Dados (UFSCAR). É pesquisador na área de Realidade Virtual desde 1995. Atualmente é docente no curso de Bacharelado em Ciência da Computação e no Programa de Pós-Graduação em Ciência da Computação do Centro Universitário Eurípides de Marília, sendo membro do Grupo de Realidade Virtual. É membro da Comissão Especial de Realidade Virtual (CERV) da Sociedade Brasileira de Computação.

Áreas de Interesse: Realidade Virtual, Realidade Virtual e Educação.
Filiação: LRV - Laboratório de Realidade Virtual - Centro Universitário Eurípides de Marília (UNIVEM).
Contato: remo@univem.edu.br

Judith Kelner recebeu seus títulos de Mestre e Doutora em Ciência da Computação do CIn UFPE, em 1981, e da University of Kent at Canterbury, Inglaterra, em 1993, respectivamente. Atualmente, é professora adjunta do CIn, onde iniciou suas atividades em 1977, e bolsista de produtividade em pesquisa do CNPq. De 1998 a 2005 exerceu o cargo de Chefe de Departamento e Vice-Diretora de Centro. É líder do GRVM e vice-líder do GPRT(Grupo de Pesquisa em Redes e Telecomunicações), e suas áreas de pesquisa incluem redes de ambiente, interfaces avançadas e realidade virtual e aumentada. Publicou 7 artigos em periódicos, 91 artigos em conferências e 4 capítulos de livro. Orientou 22 alunos de mestrado e co-orientou 1, e orientou 2 alunos de doutorado. Participou de 26 bancas examinadoras de mestrado e 8 de doutorado.
Áreas de Interesse: Realidade Virtual, Realidade Aumentada, Interfaces 3D e Comunicação Móvel.
Filiação: Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE)
Contato: jk@cin.ufpe.br

Larissa Pavarini é professora no Centro Universitário de Marília – UNIVEM nos cursos de ensino superior de Tecnologia (IST), e professora na FATEC de Garça, atuando nas áreas de computação. Obteve grau de mestre em Ciência da Computação na área de Realidade Virtual pelo Centro Universitário Eurípides de Marília (UNIVEM), em 2006. É membro do LApIS (Laboratório de Aplicações de Informática em Saúde) do UNIVEM. Outros graus acadêmicos: Bacharel em Ciência da Computação pela UNIVEM, em 2003.
Áreas de Interesse: Realidade Virtual, Realidade Misturada, Tecnologias Interativas, Educação Virtual Interativa e Multimídia.
Filiação: LApIS - Laboratório de Aplicações de Informática em Saúde do Centro Universitário Eurípides de Marília (UNIVEM) e LRV - Laboratório de Realidade Virtual do UNIVEM.
Contato: lpavarini@univem.edu.br

Liliane dos Santos Machado professora do Departamento de Informática da Universidade Federal da Paraíba. Atua junto ao LabTEVE - Laboratório de Tecnologias para o Ensino Virtual e Estatística - onde coordena projetos na área de realidade virtual. Doutora em Engenharia Elétrica pela Universidade de São Paulo (USP), mestre em Computação Aplicada pelo Instituto Nacional de Pesquisas Espaciais (INPE) e Bacharel em Ciências da Computação pela Universidade Estadual Paulista (UNESP).
Área de Interesse: Realidade Virtual, Sistemas Hápticos, Ensino Virtual, Jogos Multiplataforma
Afiliação: LabTEVE - DE/CCEN - http://www.de.ufpb.br/~labteve; Grupo de Pesquisa de Tecnologias em Sistemas Interativos e Inteligentes para Ensino Virtual Universidade Federal da Paraíba
Contato: liliane@di.ufpb.br ou http://www.di.ufpb.br/liliane

Luciano Ferreira Silva é professor contratado da Faculdade de Matemática da Universidade Federal de Uberlândia (UFU). Graduado em Matemática em 2003, obteve grau de mestre no setor de Engenharia da Computação na área de processamento da informação – Realidade Virtual pela Eng. Elétrica – UFU – MG em 2006.
Áreas de Interesse: Realidade Virtual, Educação, Matemática aplicada.
Filiação: Grupo de Realidade Virtual – UFU – Pós-graduação em Engenharia Elétrica – UFU.
Contato: luciano@mat.ufu.br e fsluciano@hotmail.com

Luís Augusto Consularo é professor do curso de Mestrado em Ciência da Computação da UNIMEP, contribuindo para a linha de pesquisa em Realidade Virtual deste curso. É doutor em Física Computacional pelo IFSC-USP, mestre e bacharel em Ciência da Computação pela UFSCar. Foi docente do Departamento de Informática na UEM-Maringá-PR, atuando no Mestrado em Ciência da Computação.
Áreas de Interesse: Visão Computacional, Reconhecimento de Padrões e Realidade Virtual e Aumentada.
Filiação: Universidade Metodista de Piracicaba - UNIMEP (Piracicaba-SP)
Contato: consularo@unimep.br

Marcelo de Paiva Guimarães é professor da Faculdade de Campo Limpo Paulista (FACCAMP) e do Centro Universitário Adventista de São Paulo (UNASP). Obteve grau de doutor na área de Engenharia de Engenharia Elétrica pela Escola Politécnica da USP-SP em 2004. Outros graus acadêmicos são: Mestre em Ciência da Computação pela Universidade Federal de São Carlos em 2000, Especialista em Sistemas de Informação pela Universidade Federal de São Carlos e Instituto Municipal de Ensino Superior de Assis em 1996 e Tecnólogo em Processamento de Dados pelo Instituto Municipal de Ensino Superior de Assis em 1994.
Áreas de Interesse: Realidade Virtual, Realidade Aumentada, e Sistemas Distribuídos e de Alto Desempenho.
Filiação: Grupo de Pesquisa em Computação da Faculdade de Campo Limpo Paulista e Centro Universitário Adventista de São Paulo
Contato: marcelodepaiva@gmail.com

Markus Endler é Bacharel em Matemática e Mestre em Informática pela PUC-Rio (1995 e 1997), e obteve os títulos de Dr.rer.nat. pela Universidade Técnica de Berlim (TU Berlin) em 1992, e de Professor livre-docente da Universidade de São Paulo em 2001. De 1994 a 2000 foi professor de tempo integral do Instituto de Matemática e Estatística da USP, onde foi membro fundador do Laboratório de Computação Paralela e Distribuída. Lá orientou 8 dissertações de mestrado e uma tese de doutorado. Desde março de 2001 é professor de tempo integral do Departamento de Informática da PUC-Rio, onde atualmente orienta 10 alunos de pós-graduação e vários alunos de graduação. Em 2003 fundou o *Laboratory for Advanced Collaboration* (LAC), que desde então está sob sua coordenação. Desde 2003 é Coordenador de Desenvolvimento do DI na PUC-Rio. É autor de um livro, co-editor de um volume da série Lecture Notes in *Computer Science*, e publicou mais de 50 artigos. É membro da SBC e da ACM, e participa de comitês técnicos de vários simpósios e workshops nacionais e internacionais na área de Redes e Sistemas Distribuídos.
Áreas de interesse: Protocolos e Sistemas Distribuídos, Computação Móvel e Ubíqua, Middlewares, Serviços baseados em localização e Colaboração entre usuários Móveis.

Filiação: Departamento de Informática da PUC-Rio
Contato: endler@inf.puc-rio.br

Nivaldi Calonego Jr é professor do curso de Ciência da Computação da UNEMAT. É doutor em Física Computacional pelo IFSC-USP, mestre e bacharel em Ciência da Computação pelo ICMC-USP. Atuou como docente do ICMC-USP, e então na UNIMEP-Piracicaba - SP, no curso de Mestrado em Ciência da Computação.
Áreas de Interesse: Realidade Virtual e Aumentada e Sistemas Distribuídos.
Filiação: Universidade Estadual de Mato-Grosso - UNEMAT (Cuiabá-MT)
Contato: nivaldi.calonegojr@gmail.com

Regina Célia Coelho é professora do curso de Mestrado em Ciência da Computação da UNIMEP, atuando na linha de pesquisa em Realidade Virtual. É doutora em Física Computacional pelo IFSC-USP, mestre em Ciência da Computação pela UFSCar e bacharel em Ciência da Computação pela UNESP Bauru. Atuou como docente do Mestrado em Ciência da Computação na UEM-Maringá-PR.
Áreas de Interesse: Computação Gráfica, Visão Computacional, Redes Neurais e Realidade Virtual e Aumentada.
Filiação: Universidade Metodista de Piracicaba - UNIMEP (Piracicaba-SP)
Contato: rccoelho@unimep.br

Ricardo Nakamura é formado em engenharia mecânica e mestre em engenharia elétrica (sistemas digitais) pela Escola Politécnica da USP (EPUSP). Atualmente dedica-se ao doutorado na área de Entretenimento Digital, participando do Laboratório de Tecnologias Interativas (Interlab), no Departamento de Computação e Sistemas Digitais da EPUSP. Já atuou como professor de graduação em disciplinas relacionadas a computação, especialmente Computação Gráfica.
Áreas de Interesse: Entretenimento Digital, Realidade Virtual, Computação Gráfica, Tecnologias Interativas Aplicadas à Educação.
Filiação: INTERLAB – Departamento de Engenharia de Computação e

Sistemas Digitais – Escola Politécnica da Universidade de São Paulo
Contato: ricardo.nakamura@poli.usp.br

Roberto Cezar Bianchini, Doutor em Engenharia da Computação, título obtido junto à Escola Politécnica da Universidade de São Paulo em 2005. Também é bacharel em Física, concluída em 1996, e Mestre em Física, título obtido em 2000. É um dos fundadores do grupo de pesquisa de jogos do Laboratório de Tecnologias Interativas da USP, onde tem participado de projetos de pesquisas e aplicação de uso de jogos no ensino e da elaboração de artigos explorando tecnologias relacionadas a jogos eletrônicos. Além de ministrar palestras sobre jogos é colunista regular da revista especializada em games *Game Master* desde outubro de 2005. Experiência docente inclui disciplinas em cursos de graduação e pós-graduação há 7 anos.
Áreas de Interesse: Inteligência Artificial, Sistemas Emergentes, Computação Gráfica, Realidade Virtual e Aumentada.
Filiação: INTERLAB – Departamento de Engenharia de Computação e Sistemas Digitais – Escola Politécnica da Universidade de São Paulo
Contato: roberto-bianchini@hotmail.com

Rodrigo Damazio é graduado em Engenharia Elétrica na Escola Politécnica da Universidade de São Paulo em 2005. Participou do Núcleo de Aprendizagem, Trabalho e Entretenimento do Laboratório de Sistemas Integráveis da Universidade de São Paulo (NATE - LSI - USP), onde desenvolveu diversos projetos na área de realidade virtual, realidade aumenta e jogos. Já ministrou disciplinas de desenvolvimento de jogos em diversos cursos e faculdades, e atualmente é engenheiro de software no Google.
Áreas de interesse: realidade virtual, realidade aumentada, jogos, jogos on-line, sistemas distribuídos e de alto desempenho.
Filiação: Centro Universitário Adventista de São Paulo
Contato: rodrigo@damazio.org

Romero Tori é Professor Ttitular do Centro Universitário Senac, onde coordena o Programa Institucional de Iniciação Científica, e Professor Associado da Escola Politécnica da Universidade de São Paulo, onde coordena o Laboratório de Tecnologias Interativas (Interlab). É Livre-

docente pela Universidade de São Paulo, na área de tecnologias interativas. É membro do Conselho Deliberativo da Escola do Futuro da USP. Foi General Chair do SVR 2004, Program Chair do SBGames 2005 e Program Chair do SVR 2007. É presidente eleito da Comissão Especial de Realidade Virtual (CERV) da SBC para o mandato 2007-2008. **Áreas de Interesse**: Realidade Virtual, Realidade Aumentada, Tecnologias Interativas, Educação Virtual Interativa e Games. **Filiação**: INTERLAB - Laboratório de Tecnologias Interativas do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo. LPAI-Grupo de Pesquisa Processos & Linguagens Emergentes: Design & Tecnologia, do Centro Universitário Senac de São Paulo.
Contato: tori@acm.org

Rosa Maria Esteves Moreira da Costa é Professora adjunta da Universidade do Estado do Rio de Janeiro. Obteve o grau doutora pela Coppe-Programa de Engenharia de Sistemas e Computação da UFRJ em 2000.
Áreas de interesse: Realidade Virtual aplicada à Medicina e à Educação; Inteligência Artificial aplicada aos ambientes virtuais e Realidade Aumentada.
Filiação: Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística, Departamento de Infromática e Ciência da Computação.
Contato: rcosta@ime.uerj.br

Sérgio Roberto Delfino concluiu o Bacharelado em Ciência da Computação pelo Centro Universitário Eurípides de Marília em 2004. Cursa mestrado em Ciência da Computação pela mesma instituição desde 2005. Atualmente é professor auxiliar da Faculdade de Tecnologia de Ourinhos e professor efetivo das Faculdades Integradas de Ourinhos.
Áreas de Interesse: Realidade Virtual, Computação Gráfica, Processamento de Imagens e Treinamento Médico.
Filiação: LApIS - Laboratório de Aplicações de Informática em Saúde. Grupo de Pesquisa em Realidade Virtual do Centro Universitário Eurípides de Marília.
Contato: srdelfino@gmail.com

Tereza Gonçalves Kirner é graduada pela Universidade Federal de São Carlos (UFSCar), com mestrado pela Universidade Federal do Rio de Janeiro (UFRJ) e doutorado pela Universidade de São Paulo (USP). Realizou pós-doutorado de 2 anos no Center for Software and Systems Engineering (CSSE) da Universidade do Colorado em Colorado Springs (UCCS), Estados Unidos, desenvolvendo trabalhos sobre engenharia de requisitos de software.
Áreas de interesse: Engenharia de Software e Sistemas de Informação, incluindo o desenvolvimento e avaliação de ambientes e sistemas de realidade virtual.
Filiação: Programa de Pós-Graduação em Ciência da Computação, Universidade Metodista de Piracicaba (UNIMEP).
Contato: tgkirner@unimep.br

Valéria Farinazzo Martins Salvador é graduada em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho (1995) e possui mestrado em Ciência da Computação pela Universidade Federal de São Carlos (2000). É doutoranda em Engenharia Elétrica na Escola Politécnica da Universidade de São Paulo, POLI/USP.
Filiação: Professora do Centro Universitário São Camilo e da Universidade Presbiteriana Mackenzie, São Paulo.
Áreas de Interesse: Engenharia de Software, Realidade Virtual e Interfaces Humano-Computador.
Contato: valeria@scamilo.edu.br

Veronica Teichrieb recebeu seus títulos de Mestre e Doutora em Ciência da Computação do CIn UFPE, em 1999 e 2004, respectivamente. Realizou o doutorado-sanduíche na empresa Aero-Sensing Radarsysteme GmbH, na Alemanha. Atualmente, é vice-líder do GRVM, aonde vem trabalhando com projetos de pesquisa e desenvolvimento. Suas áreas de pesquisa incluem realidade virtual, realidade aumentada e interação 3D. Verônica lecionou na Poli UPE, na Universidade Salgado de Oliveira e na URCAMP, entre 1996 e 2006. Publicou 1 artigo em periódico internacional, 3 capítulos de livro, e 37 artigos em conferências nacionais e internacionais. Ministrou 8 cursos de curta duração. Está co-

orientando2 alunos de doutorado, co-orientou 2 e está co-orientando outros 3 alunos de mestrado, e orientou 1 e co-orientou outros 7 trabalhos de conclusão de graduação e está co-orientando outros 2. Participou de 2 bancas de defesa de mestrado e 1 de trabalho de conclusão de graduação.
Áreas de Interesse: Realidade Virtual, Realidade Aumentada, Interação.
Filiação: Departamento de Sistemas Computacionais (DSC), Escola Politécnica de Pernambuco (Poli), Universidade de Pernambuco (UPE).
Contato: vt@cin.ufpe.br

Revisores

Cláudio Kirner
Filiação: UNIMEP / UNASP
Contato: ckirner@unimep.br

Robson Augusto Siscoutto
Filiação: UNIC / FATEC – Faculdade de Ciência e Tecnologia
Contato: robson.siscoutto@poli.usp.br e robson@unic.br

Este livro foi especialmente editado, em tiragem limitada, a partir de conteúdos desenvolvidos para o curso "Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações" apresentado no pré-simpósio, em 28 de Maio de 2007, associado ao IX Symposium on Virtual and Augmented Reality, realizado em Petrópolis – RJ de 28 a 31 de Maio de 2007, promovido pela Sociedade Brasileira de Computação e organizado pelo Laboratório Nacional de Computação Científica (LNCC).